ST ATARI
AMIGA
MAC et PC
émulation
Stations UNIX

ST MAGAZINE

Le magazine
des 16/32 bits

N° 43 / 25F
AOUT 1990

ST

ARABESQUE
RETOUCHE
GENLOCK POUR STE
MORTIMER
PROGRAMMER MIDI

Spécial Listings

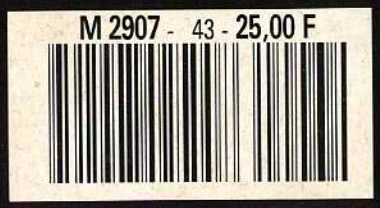
SOUNDTRACKS SUR ST
EMULATEUR MINITEL
UN JEU : DALEKS

AMIGA

AREXX ET LE GFA
LA CARTE PC

TECHNOLOGIES
AVANCÉES :

DOSSIER IRCAM



CONCOURS CREATIVITE '90
ET DEMOS : LES RESULTATS !

BELGIQUE : 180 FB CANADA : 6.95$C SUISSE : 7.50 FS

SALON INTERNATIONAL DE LA MUSIQUE 90

Du Mercredi 12 au Samedi 15
SEPTEMBRE 1990
de 11 h à 19 h
GRANDE HALLE DE LA VILLETTE
Métro Porte de Pantin

10 000 instruments de musique présentés
800 marques françaises
et étrangères.

Toute l'édition musicale.

BERNARD BECKER
BLENHEIM

22, 24 RUE DU PRESIDENT WILSON
92532 LEVALLOIS PERRET CEDEX FRANCE

EDITO

SOUS LES PALMIERS, LA PLAGE...

Excusez le clin d'oeil, mais nous ne pouvions trouver d'image plus "parlante" pour notre couverture en cette période de l'année ! Voir dans nos premières pages le résultat du Concours Créativité'90, puisqu'il s'agit du travail d'Eric Galand, un participant qui n'aura pas été primé par le Jury mais qui gagne ainsi le "prix ST Mag", et du même coup un abonnement gratuit. Encore bravo !

Comme nous vous l'annoncions le mois dernier, ce numéro est "spécial" puisqu'il est principalement dédié aux listings et à la programmation (heureusement, il n'y a pas que ça 1!...), histoire de passer des vacances studieuses. Sachant que les sujets proposés, pour certains, sont très longs à taper (mais c'est aussi un gage de qualité et d'applications surprenantes, comme l'émulateur Vidéotex), et que la chaleur n'aide pas à ce dur labeur, la disquette du journal sera cette fois, exceptionnellement, formatée en double-face. Pour ceux qui rencontreraient des difficultés liées à leur modèle de lecteur, il leur suffit de renvoyer cette disquette à la Boutique de Pressimage, en nous demandant de la transformer en deux disquettes simple-face. C'était la seule solution pour simplifier l'opération, que les intéressés veuillent bien nous en excuser.

Voilà, sinon Truc est déjà parti en vacances, Machin n'est pas encore rentré, Duschmoll a disparu de la circulation, et nous on se casse vite fait ! Bonnes vacances, vous en avez sûrement besoin, et nous encore plus...

La Rédaction

Directeur de la publication: Godefroy Giudicelli (43). Rédacteur en chef: François Gabert (42). Rédacteur en chef adjoint: Jacques Caron (44). Chefs de Rubriques: Laurent Katz (41), Stéphane Lavoisard (42), François Pagès (42), Daniel Fournier (42), Sébastien Mougey (43), François Paupert (43). 3615 STMAG: Mic Dax (43), Watsit (41 fillette). Secrétaire de rédaction: Françoise Germain (38).
Maquette et photogravure: Michel Lhopitault (42), Nicole Gabert (39).
Ont participé à ce numéro: Génaro Paz, Didier Cornuault, Thomas Conté.
Special dedicace to Hélène Sadous. Anti dedicace pour la réforme de l'orthographe, (page 146, il faut lire "sont assurés" au lieu de "est assurée", à ce propos...).
ST Magazine est une publication Pressimage, SARL de presse au capital de 2000 francs et dont le siège social est situé au 210 rue du Faubourg Saint Martin, 75010 Paris. La rédaction, quant à elle, attend vos courriers au 19 rue Hégésippe Moreau, 75018 Paris. La copie et la traduction, même partielles, sur tout support de nos textes ou documents est formellement interdite sans notre autorisation. L'envoi de textes, photos ou documents implique l'acceptation par l'auteur de leur libre publication dans le journal. Documents non retournés. Toutes les marques citées sont déposées par leurs propriétaires respectifs.
Imprimeurs: SNIL (Aulnay sous bois), RBI (Tournan). Transcodage & photocomposition: Pressimage. Photogravure couleur: Chromozone (Montrouge), STRG (Levallois), 2AGS (Tours), Pressimage (Lhopitault).
Service Promotion: Option Presse Diffusion, terminal E 13. ISSN: 0980-5338. Commission Paritaire: 71845.
Membre inscrit OJD.
Publicité: Antoine Harmel (43) et Alexandra Guillard (37) au (1) 45 22 38 60.

SOMMAIRE

GRAPHISME

Arabesque
Pierre Michel **48**

Retouche
Laurent Katz **52**

APPLICATIONS VERTICALES

Public Astro
Michel Centelles **44**

Gesbarre
Laurent Debarre **66**

MATERIEL

Les souris
Watsit & Jacques Caron **34**

Le ML68705
DiskMaster **36**

Les extensions mémoire
Pierre Michel **40**

Genlock pour STE : le GST 40E
Anne Olivelli **42**

UTILITAIRES

Twist II
François Auboux **60**

Mortimer
François Auboux **62**

BIDOUILLE

Conversion A/N-N/A
Diego Bonnani **72**

TRAVAUX PRATIQUES

Initiation au GfA (II-5)
Christophe Castro **129**

Programmation en Omikron (V)
Sébastien Enselme **138**

Initiation à l'assembleur (XVI)
C. Pascalada **125**

STOS Pratique (VIII)
Fronçois Lionnet **134**

PROGRAMMATION

GfA-GUP
Daniel Glazman **70**

Soundtracker et ST
ST Connexion **84**

Lutins et souris
Franck-Olivier Lelaidier **94**

ST Multitâche
Franck Jeannin **100**

Scrolling et GfA (VIII)
Frédéric Sannier **104**

Programmer MIDI
Stéphane Buendia **130**

Le coin propre à Billy T.
Frédéric Mora **114**

Les GfA-Punchs
Et le STOS-Punch en page 146 **142**

L'émulateur Vidéotex
Emmanuel Anne **146**

TELEMATIQUE

La rubrique Vidéotex : les aiguillages
Emmanuel Schweitzer **20**

Scandale : France Télécom
Jacques Caron & Mic Dax **22**

La rubrique des RTC
Watsit **24**

DIVERS

Abonnement
Passage obligatoire... **116**

Petites Annonces
Vendre, acheter ? **160**

La Boutique de Pressimage
Bon de commande, page 161 **25**

JEUX

Toute l'actualité...
Jean Delaite **171**

MAC

Les News
L'actualité du Mac **204**

Les ressources
Alain Raynaud **206**

AMIGA

Les News
Toutes les nouveautés **186**

La carte PC
Un pont entre deux mondes **190**

Interfacer ARexx avec le GfA
Patrick Solar **192**

Les livres sur Amiga
Patrick Solar **194**

Comprendre son Amiga
Daniel Garant & Yves Brazeau **196**

Les turmites (III)
François Fleuret **200**

Désassembleur 68000
Eric Brunet **178**

ACTUALITÉS

Créativité '90
Les résultats du concours **6**

Concours Démos
En direct de chez Gen4 **12**

Paris-cité
Agathe Sellier & Sébastien Mougey **16**

Les News
L'actualité du ST **28**

TECHNOLOGIES AVANCÉES

Cahier réalisé par Daniel Fournier.

Présentation de l'IRCAM **212**

Entretien avec J.F. Allouis **214**

Acoustique architecturale **220**

La station de travail musicale **222**

Chant & Formes **231**

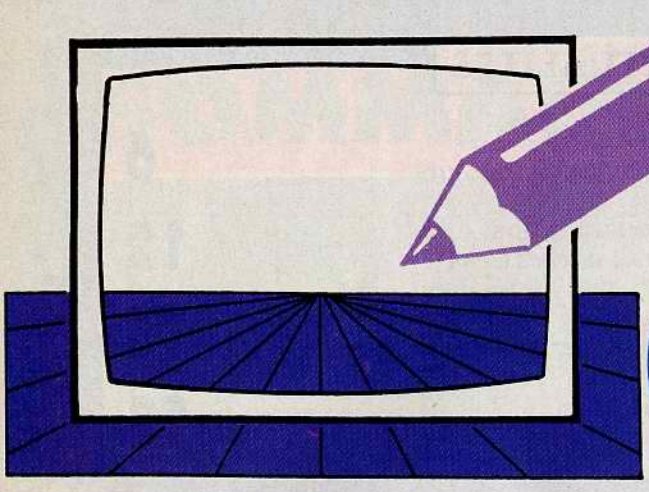
Pédagogie musicale **234**

Le Système 4X **238**

Tubes Unix (*Marc Lussin*) **240**

Index des Annonceurs

AGORA....63
ALM....55,57
AMIE....47
APPLICATION SYSTEMS....37
AROBACE....19
AXE 3D....75
BERNARD BECKER BLENHEIM....II
BONNES ADRESSES....71
CLAVIUS....33
DIGITAL CONCEPT INTER....29,31
DUTAUD....43
ELECTRON....65
ESAT SOFTWARE....45
GENERAL VIDEO....163 à 170
INFO-PROMOTION....225
JESSICO....59
KOMELEC....39
MICRO-VIDEO....68-69,99
MUST....21
RUN INFORMATIQUE....187
SARO....73
SALON DE LA MICRO....51
SCAP....27
TWENTY CENTURY SOFT....175
UPGRADE....IV
VIDEOSHOP....9,11



LE CONCOURS CRÉATIVITÉ 90

Voici donc les résultats de ce grand Concours Graphique qui aura animé nos colonnes pendant ces derniers mois, et qui aura connu une forte participation compte tenu du sujet spécifique sur lequel il portait et du travail créatif qu'il supposait. Au 1er juin dernier, date limite de participation, nous avions reçu pas moins d'une soixantaine d'inscriptions, certaines se démultipliant sur plusieurs catégories, et la catégorie "Images fixes" autorisant jusqu'à trois propositions, vous imaginez le nombre d'images à visualiser ! Une participation d'un très bon niveau global, avec quelques surprises étonnantes, le public du ST en général et le lectorat de ST Magazine en particulier faisant preuve d'un talent certain, inégal certes, mais tout à fait rassurant sur le degré d'investissement du public en matière de création graphique. Puisse la micro-informatique révéler tous ces talents dans les mois et les années à venir...

LES SPONSORS

Il nous faut tout d'abord remercier les sponsors qui ont très favorablement accueilli notre initiative et ont su mettre à notre disposition des prix prestigieux qui ont sûrement très largement contribué à la motivation des participants. Sans aucun ordre de priorité, rappelons que "NEC France" proposait un moniteur Multisync 3D, "Satellite et Télévision" un Genlock pour ST, "Digital Concept International" une extension mémoire de 4 mégas, et "Omikron France" un disque dur 32 mégas. Voilà pour le "hard", les Editions Upgrade proposant pour leur part, 4 autres grands prix non moins prestigieux puisqu'il s'agit de 4 bons d'achat d'une valeur de 3000 F TTC sur l'ensemble de leur gamme logicielle. Merci à tous, donc, d'avoir su joindre l'utile à l'agréable, en espérant trouver plus souvent des sponsors aussi attentifs à ce type de concours.

LE CONCOURS

Le Concours Créativité 90 a sollicité la participation des créatifs de toutes disciplines, des étudiants, des graphistes confirmés et amateurs, et surtout des personnes qui considèrent leur ST comme un outil indispensable à la créativité personnelle. Couvrant l'ensemble des domaines concernés par l'image de synthèse sur Atari ST, le Concours Créativité 90 s'est décliné en quatre catégories : Animation 2D/3D, Générique, Images fixes et Programmation.

L'image de notre couverture principale est l'une des images fixes proposées qui n'ont pas été primées, mais c'est un peu le "prix ST Mag"... Il s'agit d'une image Spectrum de Eric GALAND, de Tours, que nous félicitons pour son travail et dont la thématique se rapproche parfaitement de la période de sortie de ce numéro d'été de ST MAG... Quant au palmarès officiel, voir un peu plus loin !

LE JURY

Le Jury de Sélection, composé de professionnels de l'infographie, s'est réuni le 3 Juillet 1990 pour visionner les oeuvres présentées au Concours et décerner l'ensemble des prix. Voici la "fiche d'identité" des six membres du Jury :

Anne BROTOT, Fantôme. La société Fantôme (l'esprit de synthèse), spécialiste des images de synthèse en 2 et 3D, crée et réalise des spots publicitaires, des génériques, des logos, des films institutionnels, des personnages animés, pour le cinéma et la vidéo. Parmi les oeuvres signées Fantôme : Le Chien Bleu pour FR3, l'Habillage des Fêtes pour TF1, et les Fables Géométriques que vous verrez à Noël sur Canal Plus.

Pierre HÉNON, Ecole Nationale Supérieure des Arts Décoratifs (ENSAD), Atelier d'Image et d'Informatique (A.I.I.). Cet Atelier, au sein de l'ENSAD, est un organisme européen de formation initiale et continue aux nouvelles technologies de l'image. Il rassemble les compétences techniques de haut niveau d'ingénieurs informaticiens et la créativité des professionnels de l'image qui constituent son équipe.

L'A.I.I. offre une formation des étudiants de l'ENSAD avec notamment un troisième cycle de spécialisation, ainsi qu'une formation continue pour les professionnels et les entreprises grâce à des stages de courte durée permettant une approche théorique et une réelle pratique des matériels.

Thierry BRAVAIS, Mac Guff Line. La société Mac Guff Line est spécialisée dans la conception et la réalisation de films en images de synthèse tridimensionnelles, en vidéo et 35 mm. Mac Guff Line a réalisé entre autres "La Vie des Bêtes" pour Canal Plus, les génériques pour la télévision tels la Pomme d'Antenne 2, Carnet de Route pour A2, ou la Marche du Siècle pour FR3. Les images de synthèse des clips pour les Rita Mitsouko, l'Affaire Luis Trio ("Chic Planète"), Jean-Paul Gautier ("How to do that"), ou les publicités telles la montre Seiko, Orangina Light (en colloboration avec Mondino), sont aussi des réalisations signées Mac Guff Line.

Olivier WITTNER, Responsable du Département film d'animation et image de synthèse, Centre de Formation Technologique (CFT) Gobelins. Le CFT Gobelins est un établissement public d'enseignement dépendant de la Chambre de Commerce et d'Industrie de Paris, et offre une formation technologique initiale et continue liée aux métiers de l'image. Les principaux départements de l'enseignement sont : les industries graphiques, la photographie, la vidéo institutionnelle, le film d'animation traditionnel, le film d'animation en images de synthèse et l'infographie (en matière de film d'animation, 30 élèves suivent un cours sur 2 ans, tandis que les cours d'image de synthèse sont suivis par 10 élèves sur 1 an).

Isabelle PLAZANET, Thomson Digital Image. TDI est la société leader pour les systèmes de création d'images de synthèse animées 3D, et développeur de "Explore", logiciel de création d'images de synthèse de haute qualité qui tourne sur la station de travail Iris de Silicon Graphics Inc.

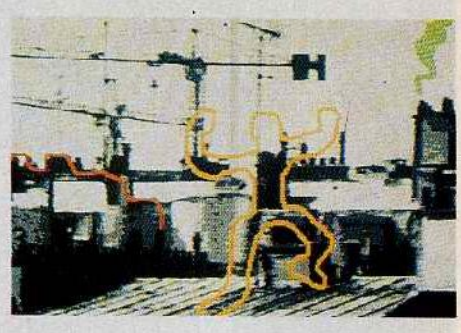
Hervé MASSERON, graphiste de son état. Il est le concepteur/dessinateur du premier film réalisé entièrement sur Atari ST, "La Clef des Songes", et faisant partie de la Sélection Officielle d'Imagina 88, le prestigieux Forum International des Nouvelles Images qui se tient tous les ans à Monte-Carlo. Auto-didacte, il travaille maintenant sur différentes stations après avoir exploité à fond les capacités du ST. Comme quoi, tous les chemins mènent à Rome...

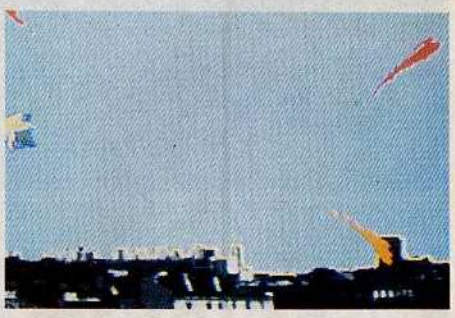
LE PALMARES

Formule originale, le Jury, après visualisation des oeuvres présentées, pré-sélection puis sélection définitive, a de lui-même proposé qu'au lieu d'attribuer arbitrairement des prix qui feraient éventuellement double usage avec des matériels déjà possédés, il soit plutôt attribué un ordre de priorité suivant lequel les gagnants se détermineraient sur le prix voulu. C'est désormais chose faite, et nous sommes en mesure de vous livrer le palmarès ci-dessous. Un certain nombre d'images accompagnent ce résultat, mais vous comprenez sans doute qu'il est très difficile, dans le cas des animations, de montrer quelque chose qui soit vraiment représentatif du mouvement. Isoler ainsi des images fixes peut même, parfois, être "dévalorisant" vis à vis du résultat animé...

Trois prix ont été décernés dans la catégorie Animation :

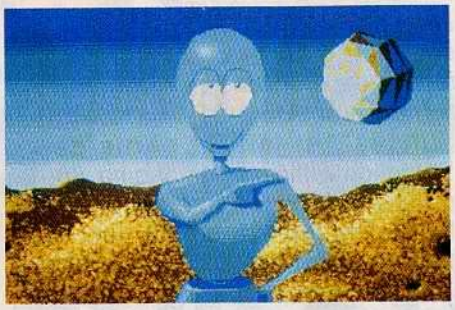
- à Mrs. ***Olivier BODEUR*** et ***Antoine MARTIN***, pour "Les Aventures de Max la Grenouille", une animation réalisée avec CyberPaint. Max la Grenouille est un personnage haut en couleurs dans une ville grise (vues de Paris en digits noir et blanc), et qui "bombe" un mur en catimini. Le motif dessiné se prend tout à coup l'envie de vivre, et va laisser des traces de couleur partout où il passe, laissant espérer qu'un jour enfin la couleur puisse prendre le pouvoir...
Max La Grenouille gagne ainsi le Genlock GST Gold de Satellite et Télévision, bravo !

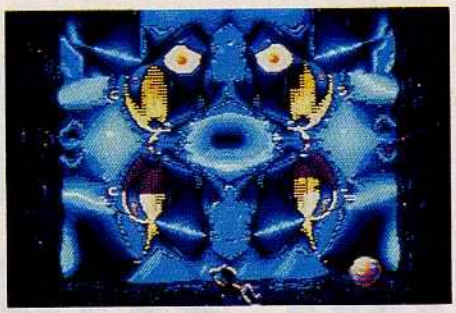




"Les aventures de Max la Grenouille" par Olivier BODEUR et Antoine MARTIN

- à Mr ***Frédéric LOCCA*** pour "Antivol", une superbe séquence réalisée à l'aide de Degas, Spectrum, CyberStudio, CyberControl, et CyberPaint. Tous ces outils ont permis à Frédéric LOCCA de générer plus de 800 images (!), avec des décors somptueux retravaillés en 512 couleurs, et dirigées par un scénario amusant où un petit être sympathique va se faire voler sa balle en 3D par un oiseau. Il le suivra jusque dans un décor dantesque où, pénétrant dans les entrailles de la terre, il se fera "foudroyé" par une divinité imposante et mécontente d'avoir été dérangée. Notre personnage et sa balle sont alors "téléportés" sur une île déserte, et lors du largage par un bizarre vaisseau, la balle tombe malheureusement à l'eau. Bonheur, une superbe raie manta viendra récupérer la balle dans les profondeurs de l'océan et la restituer à son heureux propriétaire...
Cette magnifique animation, et surtout le travail colossal qu'elle représente, sont donc récompensés par le disque dur 32 mégas offert par la société Omikron France.

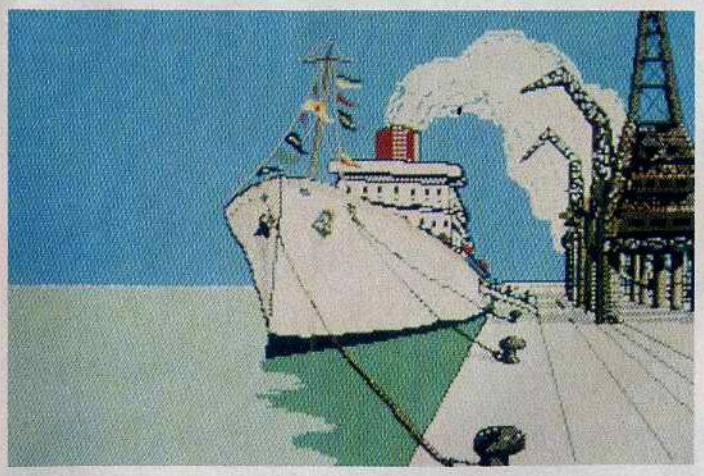


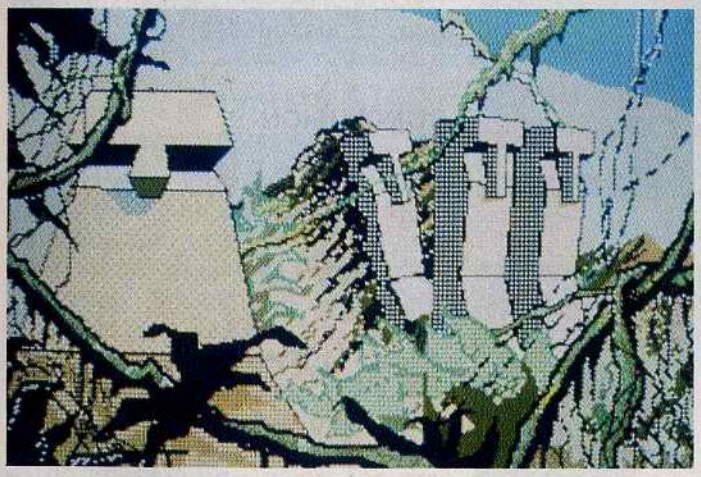


"Antivol" par Frédéric LOCCA

- à Mrs ***Frédéric BOUDET*** et ***Franck HERVÉ***, pour "Pierre contre Pierres", une séquence animée résultant d'un travail de programmation personnel et primé par le Jury pour ses graphismes originaux.
Un sculpteur en mal d'inspiration découvre, au travers d'un voyage ressemblant à une "course au trésor", la raison pour laquelle un autre sculpteur, qu'il admire et qui vit sur une île déserte, possède une source d'inspiration inépuisable. Programmée en assembleur, la routine de masques et de scrollings est un peu "répétitive" vis à vis du scénario, mais le Jury a particulièrement apprécié les graphismes. Pour cette raison, "Pierre contre Pierres" gagne l'extension mémoire de 4 mégas offerte par Digital Concept International.

Un prix a été décerné dans la catégorie Programmation, il s'agit de Mr ***Jérôme VOUILLON***, de Sassenay, qui a réalisé un travail très intéressant en basic GfA et en assembleur, en superposant deux images au for-





"Pierre contre pierres"
par Frédéric BOUDET et Franck HERVÉ

VIDEO SHOP

GROUPE ALLIANCE

3615 VS

Un micro en vacances

Du 1er juin au 31 juillet, Video Shop vous propose des offres promotionnelles "Vacances" avec au choix

PACK CADEAU *
1 manette, 50 logiciels, 1 tapis souris
+ 50 disquettes vierges avec unité centrale
+ 100 disquettes vierges avec configuration couleur

OU

30% * sur l'achat d'une imprimante ou d'un périphérique de votre choix

*Pour tout achat d'une unité centrale supérieur à 5000F hors promotion
* Promos non cumulables

GAMME 520/1040 STE

520 STE PéRITEL	3 490 F
500 STE + MONITEUR COULEUR	5 490 F
1040 STE	4 490 F
1040 STE + COULEUR	6 490 F

UNIQUE !!!
FAX à partir de 3 990 F HT
contacter Mme Blanpain
(1) 46.40.73.27

PROMO VACANCES

520 STE + Extension 2 Mo	4 990 F
520 STE + Extension 4 Mo	6 490 F
520 STE + Moniteur couleur + STAR LC10 couleur	6 990 F

SPECIAL VACANCES
Emportez un 1040 STE pour le prix d'un 520 STE 1 Mo de RAM
3490 F TTC

Tablette CRP A4	2 990 F TTC
Digitalise VIDI ST + MixImage	1 990 F TTC

Promo

Lecteur 3"1/2 DF	750 F
Disque dur 30 Mo	3 990 F
60 Mo	5 990 F
Mégafille 44	7 990 F

Logiciels

Time Works	790 F
Mega Page	990 F

NOUVEAU !!!

Le fabuleux portable d'ATARI « LE STACY »
En démonstration dans tous nos magasins.
Un outil qui concurrence à moitié prix, le portable d'APPLE
(par adjonction de l'émulateur SPECTRE CGR)

Console LYNX en démonstration dans tous nos magasins

STACY + Spectre CGR 24 900 F

Tous nos prix s'entendent TTC - GARANTIE 2 ANS
REPRISE DE VOTRE ANCIEN MATERIEL - CREDIT - LEASING - FORMATION
Catalogue contre 3 timbres à 2,20 F

POUR COMMANDER
Par téléphone : 42.86.03.44
Par Minitel : 3615 Code VS
Par courrier : Vidéo-Shop
VPC BP 105 75749 Paris cedex 15

Grand concours "Vacances"
du 1er au 31 juillet
Bulletin de participation à retirer dans chacun de nos magasins.

4 MAGASINS A VOTRE SERVICE !!!

AU CENTRE : 47 / 50, rue de Richelieu - 75001 PARIS - M° Palais-Royal
AU SUD : 251, boulevard Raspail - 75014 PARIS - Métro Raspail
A L'OUEST : 7, rue de l'église - 92200 NEUILLY - M° Pont-de-Neuilly
A L'EST : 260, rue de Charenton - 75012 PARIS - M° Daumesnil

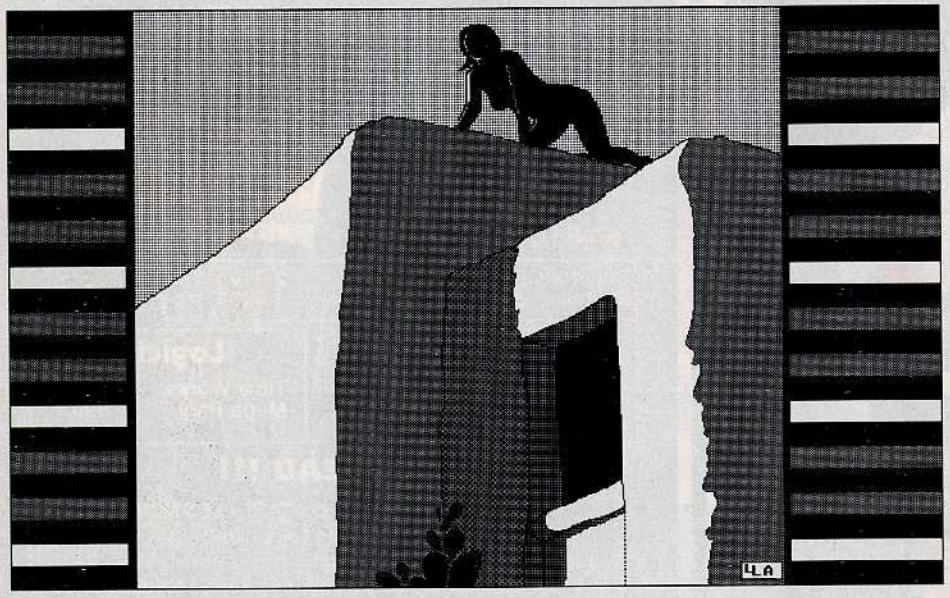
MAGASINS OUVERTS DU LUNDI AU SAMEDI
de 9 heures à 20 heures sans interruption

(1) 42.86.03.44
FAX 42.86.01.22

mat Spectrum étendu, et qui sont échangées 50 fois par seconde. Ces images (en fait, cette image puisqu'à l'oeil le "flipping" ne peut être ressenti) sont donc affichées en 24389 couleurs et le résultat est très intéressant. Bravo, donc, à Jérôme VOUILLON, qui gagne le moniteur multisync 14 pouces "3D" de la société NEC France.

Quatre Prix, enfin, ont été attribués dans la catégorie Images Fixes, et les gagnants disposeront donc chacun d'un bon d'achat de 3000 F TTC auprès des Editions Upgrade sur toute leur gamme de logiciels. Ceux-ci seront individuellement contactés afin qu'ils fassent leur choix. Il s'agit de :

"La Lune" par Jérôme VOUILLON



"PIX_ELLE" par Laurent LAFOSSE

"Création" par Sophie DEVAINON

- Mr ***LAFOSSE Laurent***, de Cannes, pour son image "Pix-Elle", réalisée en monochrome sur Degas Elite ;
- Mr. ***BERCIER David***, de Marseille, pour son image "Exether" (Degas Elite) ;
- Melle ***DEVAINON Sophie***, de La Rochelle, pour son image "Création" (Spectrum 512) ;
- Mr. ***GRÉGOIRE Jean-François***, de Savignies, pour son image "Spaguet" (Spectrum 512).

CONCLUSION

Voilà donc un chouette concours (comme on aimerait en voir plus souvent...) qui se termine de bien belle façon, et il nous faut insister sur la bonne qualité générale des oeuvres présentées. Certes, un Jury ayant à se déterminer sur la Créativité graphique ne peut être que subjectif (bien que les professionnels invités aient une grande habitude de voir des travaux de création en ce domaine) et sans aucun doute, d'aucuns trouveront leurs travaux bien supérieurs à certains de ceux qui ont été primés. Mais c'est la loi d'un Jury humain, et l'absence de récompense ne doit en aucune façon décourager tous ceux qui ont participé, les décisions étant souvent "déchirantes" dans certains cas. Bravo à tous les participants, donc, pour la preuve qu'ils nous ont donnée d'une réelle effervescence en matière de création graphique sur ST, et, nous l'espérons, "à la prochaine" !...

La Rédaction

"Exether" par David BERCIER

"Spaghetti" par Jean-François GRÉGOIRE

"Maui" par Eric GALAND

3615 ST MAGAZINE

VIDE SHOP
GROUPE ALLIANCE

Un micro en vacances

Du 1er juin au 31 juillet, Video Shop vous propose des offres promotionnelles "Vacances" avec au choix

PACK CADEAU *
1 manette, 50 logiciels, 1 tapis souris
+ 50 disquettes vierges
avec unité centrale
+ 100 disquettes vierges
avec configuration couleur

ou

30% * sur l'achat d'une imprimante ou d'un périphérique de votre choix

*Pour tout achat d'une unité centrale supérieur à 5000F hors promotion
* Promos non cumulables

MEGA ST1

Portfolio Prix vacances 1990 F avec housse gratuite !!!

	Promo Vacances
Mega ST1	4 990 F
Mega ST1 + MegaPage + moniteur mono + 1/2 journée prise en main	5 990 F
Avec Imprimante Star LC10	7 490 F
Mega ST1 + monit. mono + MegaPage + Dis. Dur 30 Mo	8 990 F
avec Imprimante Star LC10	9 990 F

Transformez votre ST en multimachine

Emulation PC		Emulation Mac	
- Supercharger	2 990 F	- Robtek	990 F
- PC Ditto II	2 490 F	- Spectre GCR + Roms	3 490 F
- PC Speed	2 490 F		

EXTENSIONS MÉMOIRE de 1 à 4 Mo à des prix fous !!!

1 Mo STE/STF	690 F
2 Mo STE	1 490 F
2 Mo MEGA ST1	2 490 F
2,5 Mo 520/1040 STF	2 490 F
3 Mo Mega ST1	3 490 F
4 Mo STE	2 490 F
4 Mo ST1	4 490 F
4 Mo STE	3 490 F

EXCEPTIONNEL !! Mega ST1 + Extension 4 Mo = 8 990 F TTC !!!

TV Pal Secam 37 cm + télécommande 1 590 F

3615 VS

Tous nos prix s'entendent TTC - GARANTIE 2 ANS
REPRISE DE VOTRE ANCIEN MATERIEL - CREDIT - LEASING - FORMATION
Catalogue contre 3 timbres à 2,20 F

POUR COMMANDER
Par téléphone : 42.86.03.44
Par Minitel : 3615 Code VS
Par courrier : Vidéo-Shop
VPC BP 105 75749 Paris cedex 15

Grand concours "Vacances"
du 1er au 31 juillet
Bulletin de participation à retirer dans chacun de nos magasins.

4 MAGASINS A VOTRE SERVICE !!!

AU CENTRE : 47 / 50, rue de Richelieu - 75001 PARIS - Mo Palais-Royal
AU SUD : 251, boulevard Raspail - 75014 PARIS - Métro Raspail
A L'OUEST : 7, rue de l'église - 92200 NEUILLY - Mo Pont-de-Neuilly
A L'EST : 260, rue de Charenton - 75012 PARIS - Mo Daumesnil

MAGASINS OUVERTS DU LUNDI AU SAMEDI
de 9 heures à 20 heures sans interruption

(1) 42.86.03.44
FAX 42.86.01.22

CONCOURS

Vous avez été 75 à participer au concours que nous organisions en collaboration avec Ashiv, distributeur exclusif de l'Archimedes en France. Il a fallu vous départager, et ce fut très dur, surtout sur ST où plusieurs groupes ont fait un boulot remarquable.

Rappelons l'objet de ce concours: vous aviez 3 mois pour réaliser une démo ayant pour thème le 3615 GEN4, le "sapristi" serveur de Génération 4. Nous avons éliminé d'emblée les démos reprenant des éléments d'autres démos ou de jeux du commerce, honte aux quelques participants qui ont osé!

Nous avons reçu des propositions venant de Belgique, du Luxembourg, d'Italie et même du Portugal, mais, bien entendu, la majorité des démos nous ont été envoyées de France.

Le niveau technique de certaines démos nous a agréablement surpris, mais il faut préciser que les démos ST étaient nettement plus impressionnantes que celles sur Amiga, compte tenu des capacités respectives des machines. De même, nous tenons à féliciter quelques graphistes, à qui nous promettons un bel avenir dans le métier!

LES VAINQUEURS

Le gagnant sur ST est un groupe bien sympathique, qui a conçu un véritable petit chef-d'oeuvre graphique. Il s'agit de... de.... Naos! Bravo à eux, leur démo est tout simplement superbe. Ils ont su réaliser 3 écrans en overscan, avec un scrolling différentiel, du multisprite, et une superbe déformation de sprite. C'est parfait!

Vous retrouverez d'autres photos de cette démo dans les pages suivantes. Elle s'appelle la "Nitrowave demo". Les créateurs de la démo sont Freddi (programmeur), Aragorn (programmeur), ATM (graphiste: chapeau!). Naos comprend d'autres membres, mais non seulement ça ne nous regarde pas, mais en plus nous n'avons pas la place de les citer tous. Saluons-les.

Sur Amiga, nous avons reçu une dizaine de très bonnes démos, dont 3 vraiment excellentes. Nous avons malgré tout réussi à en distinguer une, après quelques jours de délibération, celle de... de... Dragons!

Elle dure 12 minutes, le scénario est grandiose, les programmeurs étant allés jusqu'à inclure une musique de fin carrément larmoyante! Tous les écrans sont en overscan (facile sur Amiga), avec des vector balls, de la 3D vectorielle, des bobs, des scrolls différentiels plein écran, des dégradés, des effets de miroirs...

Félicitations à Corsair et Mister Vidéo, ils ont bien mérité leur Archimedes!

DEMOS GEN4

Naos: un écran avec beaucoup de sprites, et la trajectoire est aléatoire.

Naos: l'écran du menu (notez F1, F2, F3), un petit bijou!

Dragons: beaucoup de bobs avec des déplacements paramétrables.

Dragons: un écran en 3D vectorielle où des objets s'enchainent harmonieusement. Nous n'avons pas pu photographier l'effet miroir du bas.

Naos: Un seul motif répété sur tout l'écran, et le sprite du milieu se tord!

Les meilleures démos seront installées au cours de l'été en téléchargement, et nous réaliserons des compilations qui seront mises en vente à la Boutique à la rentrée pour un prix accessible à tous (nous les annoncerons sur le serveur au cas où elles seraient prêtes avant la rentrée).

Encore bravo aux gagnants, nous espérons que tous les participants seront sportifs, il a bien fallu faire le choix, et ça a été réellement dur!

A bientôt, peut-être?

Un grand merci à: Warning sign, Future Gang, Future Minds, S.Chanterault, G.Couvrie, STMac, The Masters, ULM, TRB, TCR (The Corporation), Gege from Tec of V8, Iron Team, HCS, The Sparks (Furyo), Dune, D.Choquez, Zuul, HMD, ST Spirit, Oya Shivo, Cakeman, Carlos Dias & Paulo Simoes, Ziggy Stardust, H.Laurent, F.Leroy, The Hammers, Equinox, JCP, Mega1, Zokathra, A.Martinet, Blue Bird, Mad Vision, Gudul, Wild Chip, Nemesis, AKC Force, S.Capo, O.Plouzennec, G.Moncaubeig, Terminators (ST), Funx, Cool Copper Crew, Snibble & Bestoff, Imagic Software, New Celeste, DDT, Spell, T.Dewavrin, Demons, Nasty Devils, Skyman & Moby, C. Vallee, MC2 & Eureka & Dragon's Blood, T.Carmona, P.Cotton, Axiome V & Leader, Tirlibibi, Digital Wonder & Graphman, Sun Connection, Bouesse, The Team, T.Landspurg, Future Mirror, Chonchon de DMA, Impact, Passion, Larcociel (DMA), Joshua, F.Azan, F.Mouthaud, S.Herremans, the Bad Xet2auq (Amiga).

ASHIV

L'importateur exclusif de l'Archimedes en France, offre à chacun des deux gagnants un A3000 accompagné de son moniteur multisynchro.

HUMAN TECHNOLOGIES

Le groupe TRB recevra un Vidi-ST, ainsi que Ziggy Stardust. Frédéric Azan recevra un Vidi-Amiga, et dix autres participants sur ST recevront un ZZ-Rough, des lots offerts par HUMAN TECHNOLOGIES, que nous remercions.

AROBACE

Un bon programmeur ne peut pas bien programmer sans son A-Debog. Arobace offre cinq A-Debog à cinq programmeurs méritants, merci AROBACE!

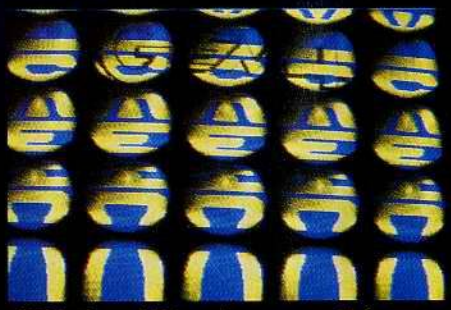
De nouvelles photos de la démo de TRB, qui gagne un VIDI-ST offert par Human Technologies! La démo de ce groupe belge comprend 9 écrans fort bien programmés, et ne comporte que des musiques soundtrack.

L'overscan et la 3D de Ziggy méritaient bien un VIDI-ST!

Deux extraits de la démo de ST Spirit, l'un inspiré de Shadow Of The Beast, et l'autre réunissant deux scrolltexts gigantesques. ST Spirit a bien mérité son A-debog!

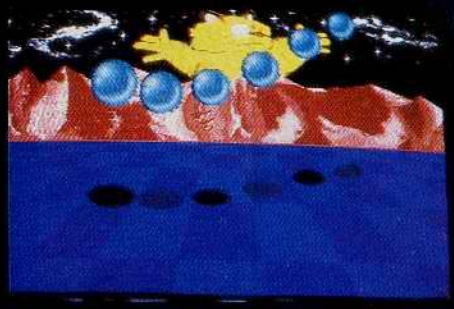
Deux écrans de la démo de The Hemoroids, qui mélange overscan et soundtrack, ce qui est tout de même une belle peformance. Ils gagnent un A-Debog.

L'énorme scroll transparent d'ULM, qui gagne aussi un A-Debog.



Deux photos extraites de la démo du groupe TEC (from V8), qui comporte 7 écrans, et qui nous a été envoyée dès les premières semaines du concours.

Un des écrans du groupe Equinox, programmé par Al Cool.



Un des écrans codés par le groupe Nemesis.

Revoici le magnifique écran de Nicolas Rougier & Patrice Borne.

Un des écrans de la démo de Dune, réalisé par Hades.

L'incroyable damier de Frédéric Azan, qui nous a vraiment laissés bouche bée. Il a bien mérité un VIDI-Amiga!

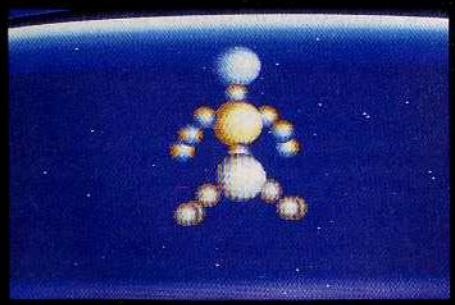
Deux photos extraites de la démo du groupe DMACON sur Amiga, qui dure plus de 7 minutes. Vous ne le voyez pas sur ces photos, mais croyez-moi: ça bouge de partout!

Les deux écrans de la démo d'Imagic, qui exploite vraiment les possibilités de l'Amiga: la boule de la deuxième photo rebondit tout en tournant!

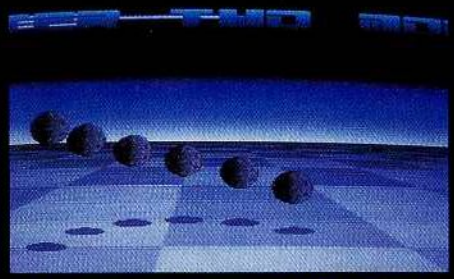
Le très bel écran d'Axiome V et Leader, tout dans le design.

Anselme et Tanguy Dewavrin nous carrément envoyé un jeu d'arcade complet, dont voici deux photos (nous n'avons pas réussi à passer le quatrième niveau)...

La démo de Demons, qui fait bouger d'énormes sprites.



Un petit bonzom en vectorballs, réalisé par Snibble et Bestof.



Le sympathique écran de MC2, Eureka, et Dragon's Blood.

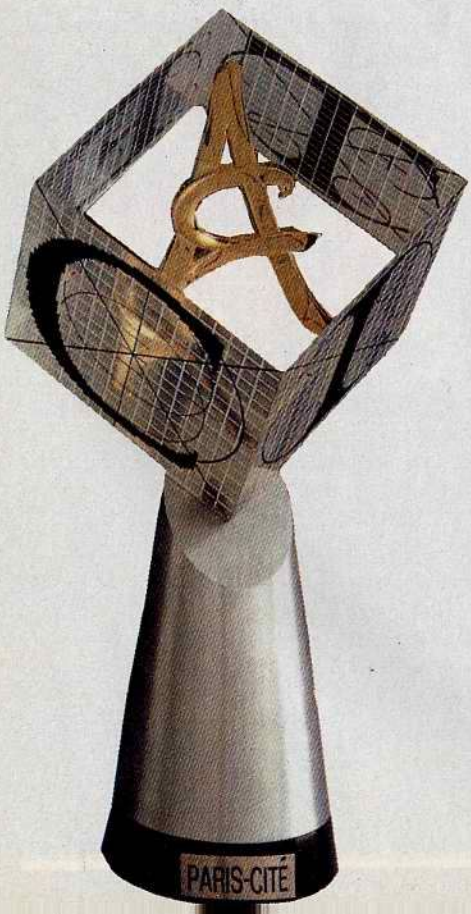
Le scrolltext en rotation réalisé par Oscar.

NOTA BENE

Certaines démos n'ont pas pu être photographiées, leurs auteurs ayant oublié de prévoir une option de gel d'écran. Dommage, il en manque de très très belles!

PARIS CITÉ'90

Grand Concours International des Technologies de la Création, ce nouveau rendez-vous des industriels et des créateurs de "tout poil" est apparu à l'initiative de la Mairie de Paris et de l'A.D.A.C. (Association pour le Développement de l'Animation Culturelle à Paris). Il a regroupé plus de quarante concours, un salon de 12.000 m2 au parc floral de Vincennes et un gala de clôture au théâtre des Champs-Élysées.



Afin de propulser Paris à l'avant-garde des technologies de la création, cette grande manifestation a mis en relation le monde des ingénieurs et des entreprises avec celui des artistes. Elle s'est articulée autour de deux axes : l'Innovation des outils et des instruments, et la Création à l'aide de ces outils. Paris-Cité a ainsi cherché à réunir des personnes de tous horizons pour mettre les nouvelles technologies au service de la culture, avec un regard neuf et efficace.

LES CONCOURS

Il s'agissait d'une compétition internationale dont l'objet est de soutenir la création artistique et l'innovation de matériels, logiciels, et instruments de création, autour des nouvelles technologies de l'image, de la musique, du langage et de l'audiovisuel. Il n'y avait "rien à gagner" au sens matériel d'un palmarès de prix offerts aux gagnants, par contre le "coup médiatique" que représente une telle manifestation constitue de fait une formidable promotion pour ceux qui ont remporté les "compétitions". Dans chaque domaine, un jury a été constitué, composé de personnalités internationales. Nous pouvons ainsi citer de nombreux représentants de grandes sociétés telles qu'I.B.M. avec Benoît Mandelbrot (créateur du nom donné aux fractales), Walt Disney Pictures, l'I.N.A. ou encore le C.N.R.S.

LA PARTIE "CRÉATION"

Divisée en quatre domaines, la Création regroupait les images (fixes ou animées), le langage (PAO, typographies,...), l'illustration sonore et l'audiovisuel (vidéogrammes infographiques). Sans vouloir présenter tous les résultats, de peur de vous lasser, voici un bref aperçu des produits les plus novateurs.

Le grand prix d'imagerie scientifique a été décerné à une visite dans une palmeraie, totalement réalisée en images de synthèse par le laboratoire de modélisation du CIRAD. D'une bonne qualité, ce produit présentait surtout un intérêt technique et esthétique, car il n'avait pas de scénario à proprement parler (ce qui n'était pas son but). Pour l'anecdote, ses concepteurs sont allés sur le terrain prendre toutes les mesures nécessaires à sa réalisation.

Sans scénario apparent non plus (ou plutôt "symbolique"), "Eurhytmy" a remporté le grand prix d'images animées. Toujours en synthèse, il s'agit d'une sorte de ballet entre des formes humaines (pour ne pas dire des hommes !), des chiens et des oiseaux qui évoluent autour d'un temple. Cette superbe animation avait d'ailleurs été présentée à Imagina'90, le festival bien connu des Nouvelles Images à Monte-Carlo.

En restant dans cette catégorie, le prix de la ville de Paris (et le prix spécial du jury en musique et son) a été attribué à "Nature morte". Cette superbe réalisation de la société RIFF associe des images réelles 35mm avec de la synthèse. Le cadre se situe dans un vieux manoir abandonné, dans lequel un couteau et une fourchette émergeant d'une longue léthargie, se mettent en demeure de goûter les fruits d'une nature morte (rappelant un peintre du XVI siècle, Arcimboldo).

Quant au prix Spécial Fiction, c'est le département Animation de la société "Fantôme" qui l'a obtenu. "La grenouille et le boeuf", tiré des Fables de La Fontaine, met en scène les "Patchitoons", personnages géométriques de dessins animés. On peut en retrouver un exemple dans un interlude de FR3, avec un petit chien jouant à la balle.

Dans la catégorie Audiovisuel, le seul produit remarquable était le prix art et fiction, avec "Réveil" de Eurocitel. Dans la même lignée que "Nature morte", soit un subtil mélange de synthèse et de réel, un réveil mécanique tente d'échapper à la traditionnelle gifle matinale du dormeur. Dans une fin tragique, il en vient à se suicider, en se jetant de la table de nuit...

Le "langage", avec sa catégorie Typographie, a récompensé une police de caractères manuscrite de François Boltana. Lors du gala, Michel Duchaussoy a lu un texte de Marivaux rédigé avec cette police, et projeté sur grand écran.

LA PARTIE INNOVATION

Elle reprend les mêmes domaines que la partie Création, mais récompense le matériel. Nous avons notamment remarqué le "Clavier rétroactif modulaire", sorte de gros champignon surmonté de touches de couleurs. Il a obtenu le grand prix de l'innovation et de la recherche musicale, ainsi que le grand prix en image-audiovisuel. Le soir du gala, son inventeur a exécuté une de ses compositions (musique contemporaine).

"Pipelette", un logiciel de l'université Paris VIII, a reçu le prix spécial du jury en image-audiovisuel. Réalisé autour d'une station Apollo Domain, ce produit simule une speakerine. Sa voix synthétique est reproduite à partir d'un texte dactylographié grâce à sa décomposition en phonèmes. De plus, l'Apollo synchronise les sons avec les images correspondant aux mouvements des lèvres et du visage. Malheureusement, la qualité d'élocution obtenue est loin d'être parfaite.

"Surface légèrement sphérique", ou le clavier rétroactif modulaire...

CONCOURS SPÉCIFIQUES

Une troisième partie comprenait des concours de création proposés par des entreprises sur leurs produits.

Ainsi ATARI France présentait le trophée STE de la création, qui avait pour but de reconnaître le travail des créateurs sur cette machine. Les gagnants sont Antoine MARTIN et Olivier BODEUR, avec les aventures de Max la grenouille en milieu urbain, animation réalisée sur Cyberpaint. Bravo pour la performance, et nous pouvons les retrouver dans notre concours Créativité'90, interne à ST Mag (mais non moins prestigieux), ce qui prouve que les grands esprits se rencontrent toujours...

Un autre concours ST, parrainé par les éditions UPGRADE, concernait des créations réalisées avec Publishing Partner Master. C'est un calendrier de V.A.V. Création qui a obtenu le premier prix.

Enfin, Thomson Digital Image (TDI) organisait un concours sur son logiciel Explore, Elison sur YSMA 18 (un réducteur de bruit), ou encore Faros sur système Simulatix.

LE SALON

Avec plus de 200 exposants, le salon du Parc Floral avait pour but de présenter les plus récentes innovations industrielles. Mais les réelles nouveautés n'étaient pas au rendez-vous, et bien que d'inhabituelles prestations aient été mise en jeu, comme la présence de bornes Sesame avec un environnement proche du Next, ou d'un plateau animé par les frères Bogdanoff, les parisiens ne se sont pas vraiment déplacés.

On pouvait y trouver un espace diffusant les oeuvres concourantes, avec neuf salles équipées de projecteurs vidéo. Le visiteur, muni d'un casque infrarouge, avait la possibilité de suivre à son rythme les différents produits. Un autre espace, le marché de l'innovation, mettait en démonstration des produits ne pouvant être projetés. Ainsi le système vidéo "Pochet" (du nom de son inventeur), qui permet une vision en relief sans lunettes grâce à un jeu de lentilles sur l'écran, était présenté.

Dans les stands, on pouvait noter la présence d'Atari France, avec un Stacy, un poste PAO et un STE interfacé avec un nouveau Genlock de la société Satellite et Télévision, spécialement développé pour le STE. Qualigraf montrait au public le NeXT, tandis que Rank Xerox réalisait chaque jour et en direct le journal du Salon.

LE GALA

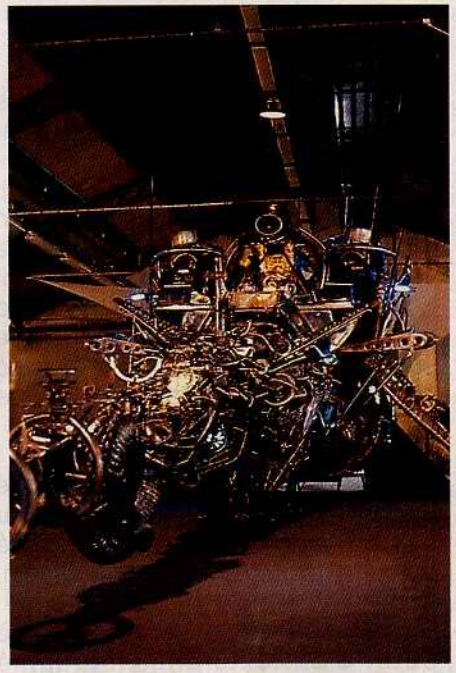
C'est dans le cadre du théâtre des Champs-Élysées qu'a eu lieu la soirée de remise des trophées. En présence de nombreuses personnalités telles que Jacques CHIRAC, Benoît MANDELBROT, ou encore le mime MARCEAU, ce gala était animé par

"Nature morte", grand prix de la ville de Paris.

Pierre TCHERNIA. Les créations récompensées étaient projetées sur écran géant dans un impressionnant décor de lasers, fumigènes, feux d'artifice et bulles de savon. Les remises de prix se sont déroulées en duplex assez farfelus avec New-York, Londres, Barcelone et Tokyo.

Voilà pour un rapide tour d'horizon de cette première manifestation, qui sera désormais biennale, mais qui n'a malheureusement, pour sa première édition, pas assez motivé les professionnels, souvent garants d'une certaine qualité. Globalement, l'ensemble de la Rédaction a été assez déçu du niveau général, avec une "mise en avant" de réalisations souvent peu novatrices ou simplistes, nous n'en prendrons pour preuve que la présentation d'une réalisation musicale essentiellement issue d'un arrangeur automatique... D'autre part, l'étiquette "internationale" était quelque peu exagérée, un manque certain de participations étrangères de haut niveau reléguant plutôt cette première manifestation au rang de Concours Régional. Mais que ces quelques regrets ne soient pas mal interprétés, l'initiative est plus que louable, l'organisation générale était plutôt bien maîtrisée malgré quelques petits ratés secondaires, et disposer d'une telle manifestation à Paris ne peut qu'apporter promotion et reconnaissance publique aux intéressés. Simplement, la cause de ce manque évident de mobilisation, tant des créateurs que du public, provient bien sûr de la jeunesse de ce nouveau rendez-vous, et il y a fort à parier qu'il acquierera ses lettres de noblesse lors de ses prochains épisodes, à condition que les industriels, les grandes sociétés de technologie, les créateurs professionnels et étrangers, s'y intéressent...

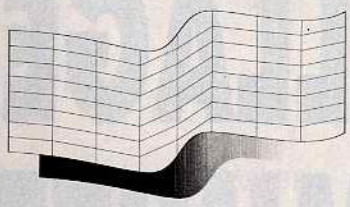
Agathe Sellier et Sébastien Mougey



Un vaisseau spatial en grandeur réelle sur le salon !!!

aroboce éditions

KSPREAD 4



Le tableur relationnel de la prochaine génération.

Multifenêtres: il permet l'ouverture en mémoire d'un nombre illimité de feuilles qui peuvent être reliées entre elles de façon dynamique.

Exploitant entièrement GEM, il présente une interface étudiée pour une ergonomie maximale.

Fonctions et macros: 132 fonctions: 14 mathématiques, 8 trigonométriques, 13 statistiques, 27 chaînes, 9 financières, 16 bases de données, 19 date & heure, 8 logiques, 24 divers.

160 macros préenregistrées. Edition de macros en ligne ou par enregistreur. L'ensemble des fonctions peuvent être sélectionnées par menus déroulants. Possibilité d'éditer ses propres fonctions.

Compatibilité: lecture et écritures des fichiers LOTUS, SYMPHONY et ASCII pour communiquer avec traitements de textes et gestion de fichiers. Ecriture de metafiles. Fonctions de traitements de zones ASCII particulièrement puissantes.

Mise en page: les possibilités de présentation sont inégalées sur un tableur: polices et taille modifiables, édition d'encadrés et de tableaux paramètrables, formatage très complet des blocs, 4 niveaux de dégradés pour les fonds, modification de la hauteur des rangées.

Graphiques: Histogrammes, lignes, graphes, camembert en 2 ou 3 dimensions. Possibilité de sélectionner des zones de données non contigües. Affichage de plusieurs graphiques à l'écran.

Impression: en mode normal ou impression graphique GDOS (compatible G+Plus), dans ce cas on obtient une qualité excellente en mode portrait ou paysage, de façon très simple. Un programme très performant de paramètrage d'imprimante est fourni.

Manuel 200 p. Prix public 990 F.

"K Spread 4 est d'une puissance fonctionnelle jamais atteinte par un tableur sur ST... son prix de vente lui procure le meilleur rapport qualité-prix ...une débauche de puissance" (Atari Magazine N 12)

""...Kspread 4 est un programme tout à fait énorme..." (ST Magazine N 41)

Renseignements produit, support technique, dialogue entre utilisateurs et avec les concepteurs

3615 AROBACE

A PARTIR D'AOUT 90

(1)42 23 50 44

GESTOCKS 90

La gestion commerciale

Facturation, gestion de stock, de comptes clients et fournisseurs.

Simple: exploitant au maximum les possibilités graphiques de l'Atari, Gestocks est très facilement exploitable même par les débutants.
Rapide: grâce à sa structure de programmation Gestocks 90 est extrêmement rapide, ce qui rend son utilisation tout à fait adaptée en tant que caisse enregistreuse.
Fiable: plus de deux années d'améliorations, des centaines d'utilisateurs, testé avec plus de 10000 produits. Aucune perte de fichiers n'a été constatée.
Modulaire: des possibilités "à la carte" peuvent être ajoutées pour des utilisations spécifiques. Support téléphonique et minitel directement avec les concepteurs.

Lecteur de codes à barres reconnu. Inventaire et statistiques multicritères. Suivi détaillé des comptes. Edition des montants TVA, du CA et des marges. Importation et exportation vers un tableur/base de données.

Prix public 1775 F

Une comptabilité générale interfaçée avec Gestocks 90 est en développement. Sa sortie est prévue pour l'Automne 90.

"...Une convivialité de ce type a rarement été atteinte sur ST...Gestocks se positionne sans problème en tête des logiciels de gestion sur ST...."(ST Magazine N 37)

"...Son extraordinaire simplicité d'utilisation, et ses possibilités de suivi des opérations en font un produit complet....(Atari Magazine N 13)"

GESBARRE

456 235

L'éditeur de codes à barres

Edite sur imprimante les étiquettes code à barres selon 7 standards existants.
4 possibilités :
-à partir d'un fichier produits Gestocks 90, de façon complète ou conditionnelle.
-en indiquant directement le message à imprimer.
-en indiquant une variable incrémentée.
-à partir d'un fichier ASCII. Par cet intermédiaire Gesbarre permet d'éditer des étiquettes sur tout fichier produit sur Atari ST ou même en provenance d'autres ordinateurs.

Utilisé en complément avec Gestocks 90 il forme le premier ensemble complet de gestion avec intégration de code barre sur Atari ST.

Il peut aussi être utilisé comme station d'impression de codes couplée à n'importe quel autre système de facturation sur d'autres ordinateurs.

Prix public 2965 F

ADEBOG

Le premier débogueur symbolique professionnel

L'outil de développement indispensable pour la mise au point de tous les programmes.
Compatible avec l'ensemble des langages compilés ou assemblés sur Atari ST (Assembleur, C, Basic GFA 3, Basic Omikron, etc.).
Entièrement en Français, il est fourni avec un Manuel de 150 pages, véritable initiation pédagogique au 68000. HotLine télématique avec les concepteurs pour les utilisateurs enregistrés.

Prix public 590 F

"ADEBOG, c'est fou ! .....Absolument génial .....sachez qu'on peut probablement tout faire, simplement, sans se fatiguer, avec ce moniteur."
(ST Magazine N 40)
"Adébog est l'aide à la programmation que tous les développeurs (professionnels ou amateurs) attendaient."
(Atari Magazine N 13)

Choisissez votre environnement

HOTWIRE 2
Lancer une application par une touche du clavier
Prix public 490 F

TURBO ST
L'affichage à la vitesse de l'éclair
Prix public 390 F

G+PLUS
L'alternative à GDOS™
Prix public 390 F

MULTIDESK
Gestion complète des accessoires
Prix public 390 F

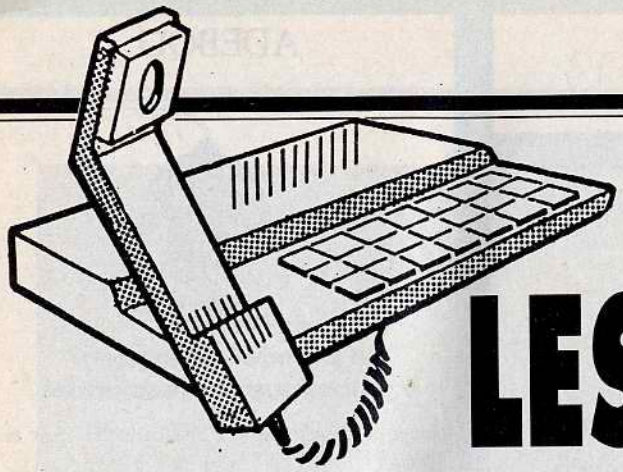
FLEXIDUMP
Le gestionnaire de copie d'écran
Prix public 395 F

CRAFT
L'environnement complet de programmation "UNIX™ like"
Manuel 600 p. 690 F

AROBACE EDITIONS
2 RUE PIEMONTESI
75018 PARIS

disponibles chez votre revendeur ou par correspondance (+ 40 F PORT)
NOM:
ADRESSE:
VILLE ET CODE:

Les marques citées sont déposées par leurs propriétaires respectifs.



LES AIGUILLAGES DU MINITEL

Les terminaux de France Telecom (les Minitels 1, 1B, 10, 12, et 2) sont bâtis autour d'une architecture modulaire découpée en 5 grands ensembles, sauf pour le M1B qui n'en comporte que quatre. Nous allons de ce pas détailler les fonctions et l'utilisation de ces fameux modules.

Voici tout d'abord une liste des cinq modules existant actuellement, accompagnés d'un rapide descriptif:

• Le module Téléphonie (qui manque à l'appel sur Minitel 1 Bistandard) sert à la composition des numéros comme on pourrait le faire depuis le clavier du Minitel.

• Le module Prise se charge des échanges avec les périphériques (dans notre cas, le micro-ordinateur).

• Le module Modem assure la transmission des données entre le serveur et le terminal.

• Le module Clavier sert à la saisie des informations.

• Le module écran assure l'affichage des informations reçues.

A sa mise en marche, ainsi qu'après une connexion ou une déconnexion, le logiciel du Minitel se charge de restaurer les aiguillages programmés d'origine, à savoir:

• Quand le modem n'est pas connecté, le clavier émet vers le modem et la prise péri-informatique, alors que le modem affiche les informations à l'écran, et émet vers lui même. La prise émet vers l'écran.
Lorsqu'on tapote sur le clavier, les caractères s'en vont par la prise vers votre ST, mais aussi au modem qui se réexpédie ceux-ci pour finalement les afficher à l'écran. C'est cette boucle du Modem sur lui-même qui génère l'écho.

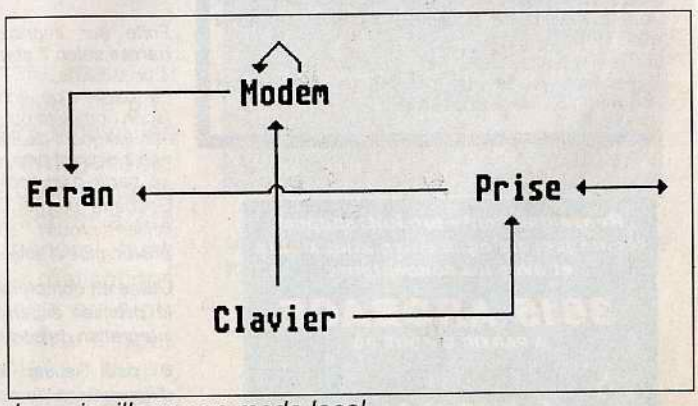
• Une fois le modem connecté, le clavier émet au modem qui transmet aux modules prise et écran, alors que la prise émet au modem. Les informations envoyées par le serveur sont donc affichées à l'écran mais aussi envoyées par la prise, qui de son côté émet vers le modem, et par conséquent vers le serveur.

Le module téléphonie n'est en principe pas connecté. Si tout cela vous paraît bien ardu, reportez-vous aux schémas joints, qui illustrent les liaisons par défaut.

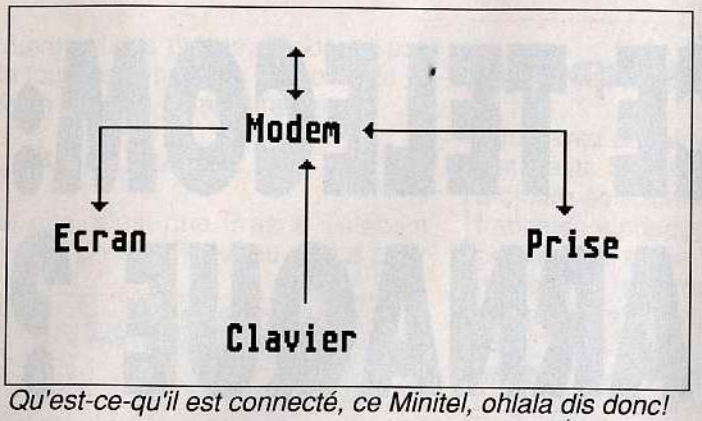
Toutes ces liaisons (qu'on appelle aussi "aiguillages") sont librement et facilement modifiables. On utilise le protocole (c'est ainsi qu'est appelé le logiciel de notre bon Minitel), qui interprètera vos commandes quel que soit l'état des aiguillages, puisqu'elles ne constituent pas à proprement parler des informations destinées au service. Elles sont donc interprétées dès leur arrivée sur le modem ou sur la prise.

Pour assurer la mise en relation des différents modules, on a affecté à chacun un code de reconnaissance qui lui est spécifique. Comme les aiguillages ne fonctionnent que dans un sens (pour avoir une liaison entre "A" et "B" il faut activer l'aiguillage de "A" à "B" mais aussi celui de "B" à "A"), il y a un code pour l'émission et un pour la réception.

La mise en route ou l'arrêt d'un aiguillage se fait en envoyant PRO 3 suivi des codes suivants: commande, module émetteur et module récepteur. La commande



Les aiguillages en mode local



Qu'est-ce-qu'il est connecté, ce Minitel, ohlala dis donc!

pour activer un aiguillage est ON ($61), et son inverse est OFF ($60). Par ailleurs, le tableau ci-dessous résume les codes de tous les modules. Ainsi, pour faire en sorte que ce qui vient du modem ne s'affiche pas à l'écran, on envoie simplement $1B $3B $60 $58 $52.

Module	Emission		Réception	
Ecran	$50	(P)	$58	(X)
Clavier	$51	(Q)	$59	(Y)
Modem	$52	(R)	$5A	(Z)
Prise	$53	(S)	$5B	([)
Téléphonie	$54	(T)	$5C	(\)

Si vous voulez utiliser ces commandes pour modifier les aiguillages du Minitel connecté (si vous êtes sysop d'un micro-serveur RTC), il ne faut pas oublier d'envoyer une demande de transparence au Minitel, sans quoi les codes seraient interprétés par votre terminal, ce qui pourrait provoquer des erreurs plus ou moins comiques. Au cas où, je vous rappelle la syntaxe: PRO2 TRANSPARENCE <x>, soit $1B $3A $66 <x>, où <x> représente le nombre de caractères que votre Minitel doit ignorer.

Voilà, c'était le dernier article de ma série. Je pense avoir fait le tour de la question (NDLR: laquelle?), mais peut-être me retrouverez-vous épisodiquement pour un article ou deux. Cependant, si vous avez des questions à me poser, j'y répondrai bien volontiers sur le...

3615 STMAG

Emmanuel SCHWEITZER,
dit STBUG.

IBM PC **serveur minitel MUST** ATARI
et Hébergements de services en 3614 ou 3615
pour CLUBS et SOCIETES à partir de 275 Fr / Mois

Sur ATARI de 1 à 8 voies

Les logiciels serveurs **MUST** sont des logiciels sous GEM conçus pour créer et exploiter facilement un multi-serveur minitel monovoie ou multivoie de haut niveau,ils fonctionnent avec un ATARI 1040 ,ou avec un deuxième lecteur,ou sur disque dur,et sur tous les types de minitel et de téléphone. Fonctionne en haute ou basse résolution :

Le serveur MUST 1 voie =2480 Francs TTC Toutes les possibillitées d'un serveur professionnel avec module de commande intégré pouvant gérer 99 catalogues de 10000 articles chacun,tout vous est possible,arborescence et pages illimitées,modifications des pages à distance,2 téléchargements TRANSTEASER,bals binaires,toutes options sysops possibles,9 niveaux possible.Le MUST des serveurs,livré en mallette avec cable détection sonnerie.

SERVEUR+KIT VIDEOTEX+25 PRG = 2990 Fr ttc

KIT VIDEOTEX 980 Francs TTC Pour ATARI 1040 ST
COMPOSEUR VIDEOTEX:Le composeur graphique professionnel sur ST **VIDEOTEXTISEUR**:des images degas...
EMULATEUR :Toutes les fonctions professionnelles,composition automatiques des N0 de tél,reception fichiers,sauvegarde,protocole de transfert incorporé,visualisation ST et minitel.**3 logiciels en 1.**
En plus 2 composeurs dynamiques 40 et 80 colonnes .

TELREP : 240 Fr ttc

Serveur accessoire ACC : Mini serveur automatique ou manuel....Lorsque votre ordinateur est occupé à une autre tache....une connexion ...TELREP prend le relais,(page minitel+dialogue minitel possible).

TELAUTO : 270 Fr ttc

Gestionnaire d'adresses ACC : composition automatique des numéros de téléphone,numéros TELETEL prédéfinis,cadran téléphonique incorporé.

Cable minitel = 140 Fr, détection sonnerie = 180 Fr

Votre serveur minitel en 3614 ou 3615

Hébergements en 3614 ou 3615
Pour clubs,sociétés,commercants à partir de 275 Fr ttc par mois

Ecran VIDEOTEX 3800 Fr ttc

Composeur de pages vidéotex **professionnel pour PC**

Logiciel pour créer un serveur TRANSPAC

HEAVEN de 4 à 128 accès sur IBM PC

TRANSPAC ou RTC 100% compatibles,avec MS-DOS
Et son prix !!! 16500 Fr Ht
En version pour RTC 2 voies = 3990 Fr ttc

☐ Je désire recevoir une documentation gratuite
NOM : **Prénom:**
Adresse :
Code postal : **Ville**
Téléphone : Revendeurs nous consulter

A retourner à : MUST B.P.004 Barneville Carteret 50270
tél : (16.) 33.53.13.66 Renseignements téléphonique
Démonstration ,commande SERVEUR MINITEL (16.) 33.04.55.55

Ou notre serveur minitel **3614 MUSTEL**

SCANDALE !

FRANCE TELECOM: L'ARNAQUE ?

Au moment où nous écrivons ces lignes, le projet de loi de Monsieur Quilès concernant la réforme des PTE, et amenant la création de deux entités autonomes appelées "La Poste" (bougez avec elle) et "France Télécom" (un avenir d'avance) a été publié au Journal Officiel, parution marquant sa mise en application. Comme vous allez pouvoir le constater, ce projet soulève quelques problèmes que nous tenons à vous exposer.

A LA PECHE AUX MOULES...

D'après les promoteurs de ce projet, il devrait permettre à ces entreprises nouvellement promues à partir de l'administration postale et des télécommunications d'affronter le marché international dans ces domaines.

... MOULES, MOULES, JE NE VEUX...

En fait, le but - souvent avoué à demi-mots, puis nié pour éviter d'offusquer - est d'imposer en douceur ce que tout un chacun refuserait, à savoir la privatisation d'un service public dont le succès a été financé par les deniers publics. Cette très probable privatisation est amenée à grands coups publicitaires depuis déjà quelques mois. Relevez le ridicule de la situation : prouver aux contribuables - à leurs frais - qu'ils ont un bon service public, pour mieux préparer son passage au privé.

... PLUS Y ALLER MAMAN, LES...

Ce projet met en évidence la boulimie de France Télécom en matière de prises de participation dans de nombreuses sociétés, jugez vous-même : Transpac contrôlé à 97 %, ou TDF à 51 % (voir encadrés). On trouve même dans le dernier rapport d'activité le titre de "Groupe France Télécom", terme particulièrement incongru dans le cadre du service public.

... GENS DE LA VILLE, VILLE...

Parlons de Transpac, justement. Rappelons que c'est la société qui gère le réseau national dit "à commutation de paquets", utilisé par Télétel (le Minitel, quoi). France Télécom utilise ce réseau pour acheminer les communications entre les Minitels et les serveurs 3615 et consorts. Chacun sait combien coûte une connexion en 3615 (1 franc la minute, pour ceux qui reviendraient d'un très long voyage). France Télécom justifie ces hauts tarifs par ceux pratiqués par Transpac (le genre "c'est pas nous c'est eux"), un discours malhonnête quand on connait les 97% cités plus haut.
Notons au passage que Transpac dispose d'un système de modulation horaire de ses tarifs (dégressifs en fonction de l'heure), qui n'est pas répercuté dans le cas des communications de type "kiosque" (3615 et plus), et ce uniquement par la volonté de France Télécom. Vous trouverez d'autres incohérences de ce type dans les encadrés ci-contre.

... VILLE, M'ONT PRIS MON...

A travers l'exemple de la fausse concurrence Alphapage-Opérator, expliqué lui aussi dans ces colonnes, on arrive au problème du monopole à outrance exercé par France Télécom, de par sa nature de service public, mais qui rivalise malgré tout, en tant qu'entreprise maintenant indépendante, avec d'autres sociétés

commerciales qui ne disposent pas de tous les avantages qui sont accordés à cette ex-administration.

... PANIER, MAMAN !

Le pire, c'est que France Télécom profite de son nouveau statut pour aller attaquer les administrations nationales d'autres pays, où la déréglementation a permis de supprimer toute forme de monopole, alors qu'en France nous restons condamnés à accepter sans broncher les exclusivités de France Télécom. Voyez à ce propos l'encadré sur l'agrément.

RENDEZ-MOI MON PANIER !

Il est relativement évident - c'est du moins ce que nous espérons - que d'autres que nous réagiront face à ce projet. En particulier, il est probable que les syndicats représentant le personnel de France Télécom agiront comme ils le jugeront bon, et cela pourra aboutir à des grèves de l'ampleur de celles qui ont déjà bloqué les Postes par le passé (rassurez-vous, ça, nous ne l'espérons pas, nous serions les premiers atteints !).

Jacques Caron et Mic Dax.

Le Minitel 2

Le Minitel, tout le monde connaît. C'est lent, c'est cher, et on peut mieux faire. Sur le principe technique, difficile de tout changer, on est encore condamné aux malheureux 1200 bauds et aux fatidiques 7 bits pour un bon moment. Par contre, les terminaux eux-mêmes sont largement améliorables, et c'est ce que France Télécom a fait avec le Minitel 2, conçu par le CNET et fabriqué par Telic Alcatel et RPIC. Enfin un terminal correct, capable de composer des numéros de téléphone, intégrant un répertoire de numéros, et agrémenté d'un mode "haute résolution", le DRCS. Tout ça pour 20 petits francs par mois. Une misère. Là où le bât blesse, c'est que France Télécom en profite ainsi pour se mettre en conformité vis-à-vis de la Cour des Comptes, qui avait jugé la distribution gratuite des Minitels comme problématique. En effet, le Minitel 2 est sorti... et le Minitel 1, le seul qui soit encore gratuit, a disparu ! Officiellement, évidemment, c'est loin d'être le cas, mais dans la pratique, il devient très très difficile d'obtenir un Minitel 1 ou 1B. C'est bien une des rares fois où l'avis de la Cour des Comptes est si rapidement respecté. Après avoir distribué gratuitement 5 millions de Minitels, France Télécom profite ainsi de l'importance prise par ce moyen de communication pour le faire payer. Tant pis pour les utilisateurs comme pour les fournisseurs de service, on change les règles du jeu en cours de partie.
Là, nous avons l'impression de tirer sur une ambulance : comment peut-on s'attaquer à France Télécom, qui a volontairement fait un manque à gagner de plusieurs dizaines de millions de francs par mois depuis presque dix ans !?

Alphapage et Operator sont dans un bateau...

Vous avez certainement remarqué le développement massif des "pagers", ces petits boîtiers ornés d'un tout aussi petit écran, et qui beepent à la moindre sollicitation. Deux concurrents semblent s'affronter sur le marché français : Alphapage de France Télécom et Operator de TDF. Si vous avez pensé une seule seconde que France Télécom a pu abandonner son monopole, retournez à la case départ, et relisez plus attentivement la partie consacrée au "Groupe".
Ainsi, vous verrez que TDF et France Télécom, c'est blanc bonnet et bonnet blanc, le premier appartenant au second (par l'intermédiaire de la Cogecom, histoire de compliquer un peu quand même). Ça explique en partie le fait que TDF puisse avoir aussi de son côté ouvert une brèche dans son monopole des télécommunications par voie hertzienne.
Et dans tout ça, c'est le contribuable qui paie encore et toujours les grandes campagnes de pub qui leur permettent de s'affronter.

Agrément

La grande majorité des utilisateurs du réseau téléphonique connait l'agrément - si ce n'était le cas, France Télécom nous en a rappelé l'existence il y a quelques mois au travers d'une campagne de publicité qui nous a encore coûté fort cher - matérialisé par la fameuse étiquette verte. Les plus débrouillards auront remarqué que cette fameuse étiquette, d'une part, est généralement synonyme de prix supérieur à un appareil d'importation non muni de cette identification, et qu'en plus, un appareil marqué de la sorte n'est pas forcément meilleur que son homologue qui en serait privé.
Disons le tout net, l'agrément n'est que pur protectionnisme, permettant à France Télécom et à ses partenaires de protéger un marché qui demeure captif. A l'heure où l'Europe approche à grands pas, on peut se demander si une telle chose peut encore exister. Eh bien, oui. D'autant plus que France Télécom, a même réussi à obtenir des autorités européennes compétentes une splendide dérogation l'autorisant à poursuivre de tels agissements, alors que la libre circulation des marchandises, et l'homogénéisation des réseaux sont à l'ordre du jour dans le reste de l'Europe ! Bref, France Télécom fait bien partie de ceux qui profitent de l'Europe, mais qui n'en pâtissent pas, aux dépends de qui ?...

50 ANS
DE TélécoMMunications INTERNATIONALES

Nous DevoNs *Eviter* d'oubLieR nos Racines Car *Plus* NouS *Progresserons* DANs le Domaine Technique, *Plus* nous Aurons *Besoin* de *Renforcer* Notre HIstoire ET NOtre *Légitimité*...

MARCEL ROULET DIRECTEUR GENERAL DE FRANCE TELECOM

La citation du Mois ! (Rapport d'activité 89 du Groupe France Télécom)

Pour 12 centimes de plus...

Les utilisateurs inconditionnels du Minitel ont tout de suite remarqué le changement : à chaque connexion sur un service, 12 centimes supplémentaires partaient dans l'escarcelle de France Télécom. Aucune raison valable n'a jamais été avancée par l'Avenir d'Avance, si ce n'est des bruits comme quoi cette surtaxe éviterait les tentatives de piratage des services, en particulier en 3615 (on ne sourit pas !). Là encore, rien n'a été négocié avec les fournisseurs de services, pourtant qualifiés de "partenaires" quand il s'agit de signer les contrats France Télécom. Ils ont effectivement signé, mais les conditions changent régulièrement entretemps, sans qu'ils en soient seulement prévenus. Un rapide calcul pour le seul mois de mai : les réseaux concernés ont enregistré 60 millions d'appels, que l'on multipliera par 0.12, ce qui dépasse les 7 (sept!) millions de francs (nouveaux!) par mois. Et c'est le chiffre minimum, puisque l'on ne connait pas le nombre de connexions réelles aux serveurs pour chaque appel : on a imaginé pour ce calcul que l'utilisateur de Télétel ne se connecte qu'à un seul serveur par appel... Qui a parlé d'ambulance?

De la notion de "service public"...

Pourquoi crions-nous au loup ? Il nous semble que notre conception du service public n'est pas compatible avec celle qu'envisage France Télécom. Le terme de "Groupe" à lui seul nous choque : pourquoi pas un holding, voire un conglomérat d'Etat ? France Télécom étant financé par l'Etat (est-il utile de rappeler que ce sont les contribuables qui remplissent les caisses ?), il a pour but de rendre service, au sens noble du terme, et non de monnayer son activité comme n'importe quel commerçant. Tout comme la SNCF ou l'EDF, l'entreprise n'a pas à dégager de bénéfices, elle devrait juste se limiter à pratiquer des tarifs établis au seul regard de la rentabilité pour des services réellement nécessaires. Payer des campagnes de publicité pour des services comme le téléphone ou les envois postaux (services "de première nécessité") pour lesquels l'Etat a l'entier monopole, est-ce vraiment nécessaire ? Payer des clips ineptes pour nous avertir que nous pouvons partir tranquillement en vacances puisque La Poste et France Télécom veillent sur tout, est-ce vraiment indispensable ? Sans compter l'effet Big Brother...
Enfin, ces campagnes de marketing ne sont pas innocentes : il s'agit de rappeler que le "Groupe France Télécom" se porte très bien, et que ce serait une excellente affaire, pour le secteur privé, que de l'acheter. En oubliant au passage que si France Télécom se porte effectivement très bien (merci pour eux), c'est d'abord et avant tout grâce au Trésor Public

LES RTC SUR ST

Avant toute chose, j'aimerais dire deux mots à tous les utilisateurs de Glopserv. Ce logiciel a l'avantage d'être absolument et très facilement paramétrable, ce qui vous permet, entre autres, de changer les pages et l'arborescence. Si je vous dis cela, c'est parce qu'il existe de plus en plus de RTC réalisés à l'aide de Glopserv, et qu'une bonne partie de ces RTC ont conservé les pages et l'arbo fournie par défaut. Cela donne naissance à toute une floppée de serveurs fort sympathiques, mais tous quasiment identiques... Des logiciels comme Glopserv, à la fois bon marché et puissants, permettent à n'importe quel passionné de télématique de créer un serveur complètement original, alors profitez-en!

iAPX va ouvrir un RTC sur ST et PC à partir du 17 Juin. Son nom: **FASHION**, au (16) 55 50 13 02, à Limoges. Deux serveurs PC et ST seront développés, un en tâche de fond et un normal pour chaque machine... Une idée très intéressante, mais je n'ai malheureusement pas réussi à joindre Fashion jusqu'ici.

DIEPTEL au (16) 35 82 44 20 devient le serveur officiel de l'ACDAS (Association Citizen-Band Dieppoise d'Assistance et de Secours). RTC donc dédié aux cibistes et radio-amateurs. Il sera parrainé bientôt par la FFCBL (Fédération Française de la Citizen Band Libre). RTC donc dédié aux amateurs de sigles bizarres aussi.

R-CULE tourne sous R-MES, fait partie du GAM, est à Marseille, a un numéro de téléphone (le (16) 91 92 37 00), vous propose des rubriques voiture et poésie, et c'est presque tout.

SOOK y'en a gentil serveur de l'ESIGELEC sous Glopserv, quoique très vide quand même. Rejoignez donc Scrib au (16) 35 62 10 65, et animez un peu son RTC!

Il y a encore des RTC sous Repteaser! La preuve, **PIXEL** en est un. Il est à Brive, au (16) 55 92 04 50, et vous propose les infos, la rubrique, les BALs et le téléchargement habituels...

LOISITEL*06 est un RTC libre, bien que ce soit le numéro 6 du réseau Loisitel. Rejoignez-le au (16) 93 71 99 07.

Le serveur **CREPUSCULE,** au (16) 51 32 96 20, vous offre des tonnes de rubriques, dont une dédiée aux Jeux de Rôles.

Il y a dans l'Essonne un RTC **MICROTEL** sous Glopserv, ouvert de 18h00 à 22h30 tous les jours et de 08h00 à minuit le week-end. Son numéro est le (1) 69 40 03 92.

Voilà, vous pouvez toujours me joindre en BAL WATSIT sur le 3615 STMAG, pour m'annoncer tout ce que vous voulez. Allez, bonnes vacances à tous les Guerriers de la Sémiotique!

RÉSUMÉ ÉTÉ 1989
CATALOGUE DE LA BOUTIQUE

Ce résumé du catalogue vous permet de connaître l'ensemble des produits disponibles à la Boutique de Pressimage. Pour en savoir plus sur chacun des produits, il faut se reporter aux catalogues parus dans les numéros d'été de ST Mag (32) et Génération 4 (13). Si vous ne les possédez pas, vous pouvez les commander au prix de 25F chacun (port compris), ou bien vous reporter au catalogue seul, que vous pouvez commander au prix de 10F (port compris), remboursable dès le premier achat.

CONNECTIQUE

CABLE MIDI 1.20m	60F
5 m	95F
COMMUTATEUR VIDEO	295F
CORDON IMPRIMANTE	145F
CORDON MINITEL	90F
CORDON PERITEL	260F
RALLONGE JOYSTICK/SOURIS	
-20 cm	60F
-2 m	95F

DISQUETTES ET LECTEURS

DISQETTES TDK SF DD	
- la boîte de 10	100F
LECTEUR SF 354	800F
LECTEUR DF (Pour anciens 520 ST)	990F
RANGEMENTS TOILES	
-10 disks	120F
-20 disks	160F

HOUSSES ET SOURIS

HOUSSES ST et AMIGA	
-écran	65 à 85F
-clavier	85 à 105F
MOUSE MAT	65F
TRACK BALL 3.0	345F

DIVERS

RUBAN SMM804	60F
RUBAN STAR NL-10	95F
CASSETTES AUDIO	
-lot de 10	30F
FILTRE DIGIT DG88	2200F

LIVRES

- ST	
BIEN DEBUTER SUR ST	129F
GRAPHISMES EN 3D	179F
GRAPHISMES EN GFA	249F
TRUCS ET ASTUCES en Gfa	269F
MUSIQUE ET MIDI	149F
SOS GFA BASIC	149F
AU COEUR DE L'ATARI ST	90F
- AMIGA	
BIEN DEBUTER SUR AMIGA	149F
LE LIVRE DE l'AMIGA BASIC	249F
LE LIVRE DU GRAPHISME	249F
- DIVERS	
COPILOT	145F
INITIATION AU Gfa (fiches seules	75F
- avec classeur	100F

LOGICIELS ATARI

ANCIENS NUMEROS

ST MAGAZINE (depuis No 3)	25F
GENERATION 4 (depuis No 1)	25F
MICRO IMPRESSION (depuis No 1)	25F

LES DISQUETTES DU JOURNAL

Retrouvez tous les listings publiés dans ST MAGAZINE sous forme de fichiers .DOC ou directement sous forme éxécutable.
Alors fini la corvée de la saisie !!
l'unité 75F

Achats depuis l'étranger

Pour les paiements sur l'étranger nous ne pouvons accepter que trois modes de règlement .

- Le **mandat postal.**
- **L'Eurochèque** avec une majoration de 71.16 FF sur le montant de votre facture.
- **Le virement "SWIFT"** de banque à banque :

Pour le compte de la société **Pressimage**

Compte numéro : *20252303*
Banque : *SG Louis Blanc*
Code guichet : *03500*
Code "SWIFT" : *SOGE FR PP*

En **N'OUBLIANT PAS DE PRECISER A VOTRE BANQUE LE MOTIF** de la transaction, par exemple : abonnement, achat disquette..

JEUX ET LOISIRS

ARK-ED	75F
ARK-ED II	75F
ASTROLOGIE	195F
ATOMIA	75F
AUTO ECOLE	95F
BATAILLE NAVALE	75F
BREAK YOUR MIND	75F
CYBERTRON	75F
DEMO BOLO	75F
DEPASOFT	75F
GAGS	75F
JEUKRAK	75F
LES GROSSES TETES	75F
LOTO	75F
MAGICPACK	195F
MEC LOVE STORY	75F
MEMORIX	75F
MINOS	75F
MONI MORSE	75F
MONOPOLY	75F
MOTS CROISES	145F
POKER CLUB	75F
PUZZLES	75F
PYRAMINOS	75F
QI TEST	75F
QUIZZ	75F
RESEAUX NEURONAUX	95F
REVERSI BRAIN (couleur)	95F
REVERSI BRAIN (toutes résol.)	145F
SIMUL GESTION ENTREPRISES	75F
SNARK I	75F
SPACE KILLER	75F
SPACE TILE	75F
SPOOK	75F
ST PORTRAIT	75F
STRIP BREAK-OUT	75F
UMS Scenary Disk ANTIQUITE I	75F
X MASTER	75F
YAMS	75F

GRAPHISME

AQUARELLES	75F
ATADRAW	195F
CLIPBOARD	75F
DEGAS COLLECTION 1	75F
DEGAS COLLECTION 2	75F
DEGAS COLLECTION 3	75F
DEMO IMAGIC I	75F
DEMO IMAGIC II	75F
ECRAN AMIGA	75F
GEMFED	75F
GRAPHIC DEMO	75F
FIXIMAGE	95F
IMAGES DIGITALES	75F
LE DEFOULOIR	75F

LES NOUVEAUTÉS

VIRUS KILLER	20F
ATABASE	250F
8 AMERICAIN	75F
PUNCHS Vol. 1	75F
COCKTAILS	95F
SHAFTESBURY	75F
MATERNELLE 4/5	95F
BACKUP ST	250F
HYPER ST	195F
AWELE	75F
PHARM-ASSIST	75F
MATHS-ATARI	75F
MEGABANK II	250F
M1 MIDI EDITOR	195F
3D MOLECULE	250F
SIGN GEM	75F
B.S.U.	75F
SPECIAL Scrolling	75F
COMM 03	250F
MATERNELLE 5/6	95F
HERSHEL	250F
S950 MIDI PROG.	195F
PIXEL HUNTER	250F

SPECLOAD	195F
TINYCLIP	195F
TINY COLLECTION 1	75F
TINY COLLECTION 2	75F
TINY COLLECTION 3	75F
UTILITAIRES DEGAS	100F
ZZ-ROUGH 1.0	195F
LES TAHITIENNES	50F
LES ANNAMITES	50F
LES STARS DU X	50F
GROS PLAN	50F

COMMUNICATION

COMMUNICATION 1	75F
CYRUS	550F
KERBIT	195F
ST COMM	75F
ST COMPO MONOCHROME	200F

LA PROGRAMMATION

BASALG	150F
CREER UN JEU EN GFA	75F
GEM KIT	95F
GFA-TINY	75F
HELP 68000	75F
LIBRAIRIE ASSEMBLEUR	75F
LIBRAIRIE OMIKRON I	75F
LIBRAIRIE OMIKRON II	75F
LIBRAIRIE PASCAL OSS	75F
LOGO Francais BASIC Corrige	75F
OSCAR OMIKRON	75F
SOURCES C	75F
SOURCES PASCAL	75F
STOOLS	75F
KIT GFA-LINK - ST GEM	195F
KIT GFA-LINK - ST MASTER	195F
KIT GFA-LINK - ST MATH-STAT	195F
KIT GFA-LINK - ST MATHS	195F
ST GEM	75F
ST MATHS	75F
ST MASTER	75F
SUPER PACK	145F

LE COIN MEDICAL

Vous y trouverez une multitude de sujets médicaux, en passant par les planches du bassin en coupe, les tablaux pharmacologiques usuels ainsi que des thérapeutiques de psychiatrie, homéopathie ...

ORDONEWS-AIDE.ACC	95F
ORDONEWS 1: DERMATOLOGIE	95F
ORDONEWS 2: O.R.L.	95F
ORDONEWS 3: DIETETIQUE	95F
ORDONEWS 4: PSYCHIATRIE NEUROLOGIE	95F
ORDONEWS 5: CARDIOLOGIE RHUMATOLOGIE	95F
ORDONEWS 6: HOMEOPATHIE	120F
ORDONEWS: TOUT!	450F
ORDONEWS: DEMO MEDI-ST	95F
ORDONEWS: MEMOS.ACC	120F
ORDONEWS: MAGAZINE.ACC	150F
ST-DIET 1	95F
ST-EPID 1	95F
ST-BIO 1	95F
ST-PHARM 1	95F
ST-ANAT 1	95F

MUSIQUE ET SON

ANIMAUX 1	75F
ANIMAUX 2	75F
CREER LE SON EN GFA	95F
CZ PHONIX	75F
EDITEUR ROLAND D10	195F
INTERMUSIC	195F
FB01 TERMINATOR	195F
INTER-DIGIT	195F
MIXTABLE	95F
MUSIQUE MAESTRO	95F
MYREPLAY	95F
MT 32 EXTENSION	195F
NATURE 1	75F
NATURE 2	75F
OCEAN 1	75F
OCEAN 2	75F
TED	195F
TIR/EXPLOSION	75F

VOTRE CERVEAU NOUS INTERESSE !

Vous pouvez commercialiser vos logiciels par l'intermédiaire de la Boutique. Il suffit de nous envoyer votre programme sur disquette. Si celui-ci est retenu, vous toucherez environ 26% du prix de vente hors taxes (contactez nous pour de plus amples renseignements). Par pitié, là aussi, ne nous écrivez pas en décrivant votre logiciel pour savoir s'il nous intéresse ou non. Tous les projets sont bons, c'est la réalisation qu'il faut juger ! Envoyez-le directement, car nous ne pouvons répondre à votre courrier. Un délai d'un mois minimun est à prendre en compte pour que le service **Collaboration Boutique** vous renvoie des éléments, car les propositions sont très nombreuses. Contact privilégié pour les propositions et le suivi des auteurs:

"Collaboration Boutique"
PRESSIMAGE
210 rue du Faubourg St Martin
75010 PARIS

AVEZ-VOUS LA COLLECTION COMPLETE DE ST MAG ???

LE PACK DU SIECLE N°1

Comprend :

4 Reliures ou Coffrets + (n'oubliez pas de le préciser) la collection complète de ST MAG du N°3 au N°30 (soit 28 numéros) bande de veinards, pour seulement **650 F** (port compris)

NOUVEAUTÉS DU MOIS

DISPAC	20F
Joueur de Cartes	75F
Assembleur n° 2	75F

L'AVENTURIER FOU

Chaque volume contient les fichiers de solutions de célèbres jeux d'aventures comme Guild of Thieves ou King Quest.

AVENTURIER FOU 1	75F
AVENTURIER FOU 2	75F
AVENTURIER FOU 3	75F
AVENTURIER FOU 4	75F
AVENTURIER FOU 5	75F

PEDAGOGIE

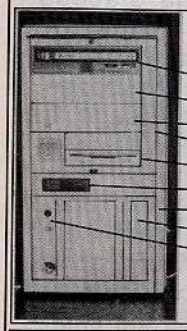
CALCUL CE ET CM	75F
CHEZ LE MARCHAND	195F
CONNAITRE LA FRANCE	145F
GEOMONDE	95F
ORTHOGRAPHE par le DESSIN	75F
SOLFEGE	75F

UTILITAIRES

ACC. DE BUREAU VOL 1	75F
ACC. DE BUREAU VOL 2	75F
ACC. DE BUREAU VOL 3	75F
ACCLOAD	75F
ASTROLAB	145F
ATHENA	295F
AUTOMATE	75F
CHECK DISK	100F
CLUSFAT	95F
GENIALOGIES	195F
GESFAM II	195F
GFA SHELL	75F
GUTEMBERG	195F
IBM-ST DISK	95F
ICONES	75F
IMPORT-PP	95F
INITPATH	145F
JOSHUA'S UTILITIES	75F
L'ETUDIANT	195F
MEGAKEY	145F
PLOT IT	145F
RAM DISQUES	75F
SUPER FORMATEUR	75F
SUPER SELECTEUR	95F
THE EXTENDER	75F
TOUCHES MORTES	75F
UTILE V2.00	95F

PROMOTIONS D'ÉTÉ SUR TOUT NOTRE STOCK

LE SPECIALISTE AU NORD DE PARIS



MEGA TOWER ALM

- Disque dur 44Mo amovible
- Disque dur 30Mo
- Emulateur PC SuperCharger
- Emulateur Spectre GCR
- Lecteur 1,44 Mo
- Hypercache 16Mhz
- Carte grand écran 19"
- Ecran Multisyncro
- Reset en façade, clavier RTS...

Reprise de votre matériel
Système modulable et évolutif

PROMO DU MOIS

Lecteurs externes complets 3"1/2, double face
750,00 frs

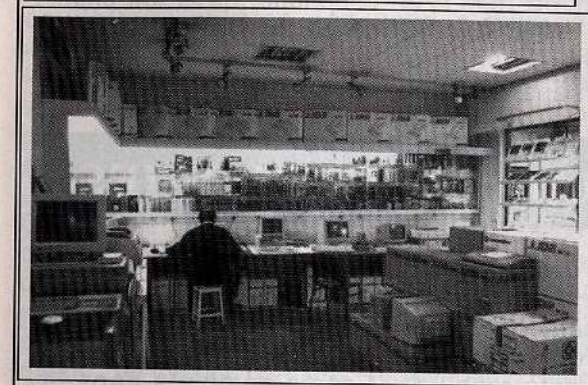
EXTENSIONS RAM

Extension 520 STE à 1MO
590,00
Extension 520 STE à 2Mo
1690,00
Extension 520 STF à 1Mo
950,00
Ces extensions sont montées par notre SAV

SUPERCHARGER

L'Emulation PC que tout le monde attendait. La vitesse d'un XT à 12Mhz, un boîtier externe de très belle qualité ne nécessitant aucune soudure (connexion sur le port DMA sans monopolisation), 512Ko de RAM (extensible à 1Mo), supporte le coprocesseur 8087, émulation CGA, Hercules, livré avec DOS 4.01, gère les disques durs Atari, le port parallèle à 100%, le port série, la souris Atari. Indice Norton 4.2...

2590 F (512Ko) – **2890 F** (1Mo)



Près de 100Mo pour 9390 frs

Disque dur Megafile 44
Livré avec 2 cartouches de 44Mo

IMPRIMANTE

Star LC 10
Star LC 10 couleur
Star LC 24-10
Epson LQ-500
PROMO !!!

INTERESSANT

Moniteur Multi résolutions
Couleur et monochrome
4490,00 frs
(reprise de vos moniteurs..nc)
Livré avec son interface.

OFFRE P.A.O.

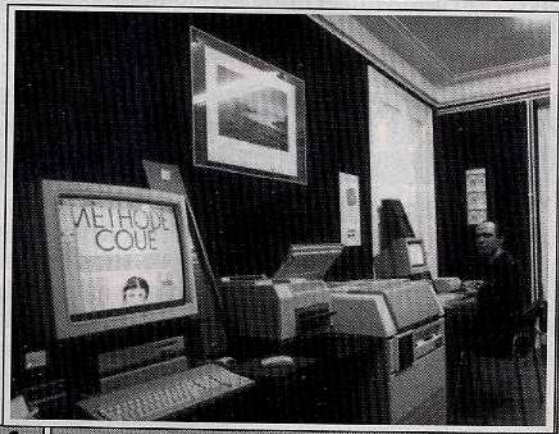
ATARI MEGA ST4
Ecran monochrome
Disque dur 30Mo
Imprimante laser Atari
Calamus, Le Rédacteur,
Timeworks, formation
Maintenance sur site
Assistance téléphonique
35000,00 frsttc

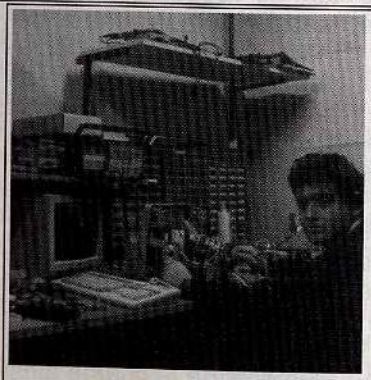
S.A.V. EXPRESS

L'événement de l'année :
1- Réparation sous 48H (Tarif normal)
2- Réparation immédiate (Tarif express)
Finies les attentes interminables !..

FLASHAGE CALAMUS

Service d'impression très haute résolution de vos fichiers Calamus sur Linotronic 300. Demandez notre service flashage.





NOUVEAU

Pour CALAMUS
Générateur de drivers pour HP et DOT MATRIX

OCCASIONS

1ère main des machines révisées garanties 6 mois à des prix défiant toute concurrence
Appelez-nous

SCANNERS A4

200dpi, 16 niveaux de gris
4990,00 frs
600dpi, 64 niveaux de gris
15200,00 frs

VOTRE ST TURBOCOMPRESSÉ

HYPERCACHE

Vous avez désormais la possibilité de doubler la vitesse de votre Atari pour très peu de frais. Un 68000 à 16Mhz et 8Ko de mémoire cache vous apportent un gain de temps pouvant aller jusqu'à 70% suivant les applications. 100% compatible ST et Mega ST. Livré avec un manuel complet en français (possibilité de montage par notre service technique)

NOUVEAU PRIX 2490,00 frs

PROMO ARCHIMEDES

Modèle A 3000 — **7990,00 TTC**
Modèle A 3000 + moniteur coul. — **9990,00 TTC**
Modèle A310...11850,00/Modèle A410...15990,00

NOUVEAU

Lecteur 1,44Mo, interne ou externe pour votre ST entièrement compatible !

L'ATARI SUR RÉSEAU ETHERNET

Réseau Biodata en démonstration permanente.
Connection entre PC, ST, UNIX...
Vitesse de transfert 10MBits/s, connection sur le port DMA, partage des ressources (disque dur...).
Possibilités de prêts/démonstrations sur site

SUPER

Reprise aux meilleures conditions de votre ST pour tout achat d'un MEGA ST ou d'un STE
PROMOTION

DOMAINE PUBLIC

LE NOUVEAU CATALOGUE 1990 POUR ATARI EST ARRIVÉ

Tous les DP de RFA,USA,GB pour notre nouveau cru
Envoyez-nous 25F en timbres pour le recevoir
30 frs la disquette, la 5ème gratuite !!!

Document entièrement réalisé avec calamus et flashé sur linotronic - Prix succeptibles d'être modifiés sans préavis - Promotions non cumulables

S.C.A.P. INFORMATIQUE

62, rue Gabriel Péri – 93200 Saint-Denis
Métro Saint-Denis Basilique–Téléphone: 42.43.22.78 – Téléconie: 42.43.92.70

AT SPEED

La course à l'émulation PC et à ses performances continue, voici qu'apparaît sur le marché français "AT Speed", réalisé par les concepteurs de PC Speed, et annoncé pour la fin du mois de Juillet en disponibilité commerciale. Comme son nom l'indique, il est censé rejoindre les performances d'un PC AT, avec un indice Norton annoncé de 6.1, soit une rapidité 6 fois supérieure à celle d'un PC XT cadencé à 4,77 MHz. Il est compatible avec Windows 3, gère toujours les 704K du MS-DOS, et surtout il émule désormais les modes graphiques Hercules, CGA, Olivetti et Tandy 1000. La carte fait la même taille (90x90mm), possède toujours son NecV30 à 8 MHz mais avec un petit nouveau, le 80286 qui est, lui aussi, cadencé à 8 MHz. Par contre, il est destiné à être installé sur le bus d'extension du 68000, ce qui signifie qu'il ne pourra fonctionner que sur les Mega ST. Sinon, il faut signaler que son logiciel de lancement fonctionnera comme un accessoire de bureau sur ST, ce qui permettra de passer d'un mode à l'autre, soit de ST à PC puis revenir au ST en retrouvant la configuration telle qu'on l'avait laissée sous ST. AT Speed émule aussi les souris Atari et Microsoft, gère à 100% le port parallèle, est compatible avec le Blitter, et devrait coûter un tout petit peu moins de 3000 F. Voilà, vous en savez autant que nous, rendez-vous sans doute en Septembre pour un essai complet de la "chose"...

A.L.M. TOUJOURS

Décidément, il se passe beaucoup de choses grâce à l'intérêt que porte cette société à l'Allemagne et à son marché sur ST. Deux événements de taille polariseront sans doute l'attention à la rentrée, à savoir l'import en France de la carte d'acquisition vidéo "Chili", de chez Marvin AG, et du logiciel de retouche d'images "ReproStudio", deux produits dont nous vous avions déjà dit deux mots dans notre compte-rendu du CeBit d'Hanovre en Mars dernier. La carte Chili possède la particularité d'être à la fois un digitaliseur, un genlock, une carte graphique, un générateur d'effets vidéo et de titrage, et un correcteur d'images (dessin et couleurs). Elle permet d'obtenir 256 couleurs parmi 65.536, propose pas moins de 6 modes de résolution écran mais se branche sur le bus d'extension du 68000 donc ne fonctionne que sur les Mega ST. Voilà pour les caractéristiques générales, et signalons tout de même qu'il s'agit de la première vraie solution en matière d'acquisition vidéo temps réel pour les ST. Nous y reviendrons évidemment en détail dès que le produit sera disponible.

ReproStudio est quant à lui un logiciel de retouche d'images, encore actuellement en phase de francisation et de finalisation, mais devrait être disponible début Septembre en version Junior, puis quelques temps plus tard en version Pro. Signalons là aussi, comme tous les produits de cette même famille, qu'il tire sa pleine puissance de la présence d'une carte graphique (pour plus de couleurs - 256 simultanément- et de résolution écran) et qu'il est un des rares logiciels développés sur ST à être compatible avec la carte Matrix bientôt distribuée par Human Technologies (voir quelque part dans ces pages de news...). La version junior devrait coûter aux alentours de 1500 F, et aucun prix n'est encore avancé pour la version Pro. C'est sans doute le logiciel de retouche qui dispose du plus grand nombre de fonctions graphiques et d'outils de dessin, avec les courbes de Bézier notamment, et une interface facile d'accès avec un grand nombre de fonctions de retouche sur les couleurs, le masquage, l'estompage, le lissage, le réglage du constraste et de la luminosité. Bref, beaucoup d'activité chez ALM pour la rentrée... Signalons d'ailleurs, après vos nombreux coups de téléphone, qu'ils ont changé de numéro de téléphone, et voici le nouveau : 42.43.36.95.

HUMAN TECHNOLOGIES

Disponibles dès ce mois de juillet, voici deux gammes de cartes haute résolution couleur, déjà évoquées dans le compte-rendu du salon de Hanovre : tout d'abord des cartes 640x400 en 256 couleurs, 800x600 en 256 couleurs jusqu'aux cartes 1280x960 en 16 couleurs. La palette s'étend sur 262.144 couleurs soit 64 nuances par couleur de base (rouge, vert, bleu). Ces cartes sont particulièrement destinées aux écrans multisynchrones 14 et 16 pouces de type EIZO, NEC ou équivalents. On trouve ensuite les cartes pour moniteurs 19, 20 ou 21 pouces avec des résolutions de 1280x1024 en 16 couleurs et de 1280x1024 en 256 couleurs parmi 16 millions. Toutes ces cartes étant disponibles seules ou avec écran, les prix s'étalent de 6900 F HT à plus de 40000 F HT, en fonction de la résolution et de la taille de l'écran choisies. Essais complets dans le prochain numéro.Ensuite, ils nous ont développé un commutateur électronique d'imprimante, proposé au tarif de 1990 F HT, permettant de partager une imprimante laser SLM804 entre deux unités centrales, sans perturber le bus DMA ni les disques durs connectés. Le boîtier permet de commuter, à la volée et sans précautions particulières, l'imprimante laser associée vers l'un ou l'autre des postes sans avoir à réinitialiser l'un quelconque des éléments de la chaîne ainsi formée. Les deux postes ne doivent cependant pas être éloignés de plus de 6 mètres pour un fonctionnement totalement fiable.

36 15 ST MAG

LE MODEM CAP 23

Le CAP 23 est un modem mis au point par la société ExtraDOS. Son but, étant conforme à l'avis V23 du CCITT, est de venir remplacer votre Minitel qui est, comme chacun sait, cher, encombrant et laid. En fait, on ne voit pas trop quel autre usage on pourrait en faire, étant donné que la norme V23 n'est utilisée qu'en France. Ainsi, si vous utilisez régulièrement des émulateurs pour ne pas avoir à supporter l'écran et le clavier du Minitel, ou si vous vous servez de ce dernier uniquement pour son modem (si vous possédez un RTC, par exemple), le CAP 23 serait susceptible de vous intéresser. Dans ces deux cas de figure, il est en effet inutile de s'encombrer du Minitel lui-même : un modem peut très bien le remplacer.Le CAP 23 se présente sous la forme d'une cartouche, à insérer dans le port adéquat de votre ST. Voilà une idée originale, qui permet probablement de réduire sensiblement les coûts de fabrication ; cela se ressent sur le prix de vente du CAP 23, qui est nettement inférieur à celui des modems V23 externes classiques. Notez cependant que cela fait du CAP 23 un périphérique exclusivement réservé au ST : inutile d'espérer vous en servir si vous passez un jour sur Mac (ou sur NeXT, PC, Sun, ce que vous voulez). Une fois votre modem-cartouche dûment inséré, il vous faut placer dans votre dossier AUTO un petit programme qui va détourner vers la cartouche les entrées/sorties normalement destinées au port série. Pour ceux que cela intéresse, les fonctions BIOS détournées sont Bconout et Bconin (les pointeurs sur les routines chargées du port série sont dans les variables système), et une petite routine est insérée dans le Timer A, afin de prévenir le système de l'arrivée d'un caractère sur le port cartouche devenu série, si besoin est. Une fois le programme AUTO installé, et le ST rebooté, tous les logiciels que vous lancerez seront persuadés qu'ils se servent du port série, à moins qu'ils n'utilisent eux aussi le Timer A, auquel cas quelques problèmes de cohabitation peuvent peut-être se poser. D'autre part, un accessoire permet à tout moment de remettre en place la RS232 standard, par exemple si vous avez un autre modem branché dessus (ce qui est mon cas).Le modem a été essayé avec divers logiciels de communication, qui se sont tous laissés berner sans aucun problème. ZZ-Comm émule comme d'habitude, Kermit transfère des fichiers sans broncher, et Uniterm supporte toujours aussi mal le Vidéotex. Le CAP 23 dispose, de plus, de quelques facilités intéressantes pour les sysops de RTC : il est notamment parfaitement conforme à la norme mise en place par le Minitel 2 concernant la détection de sonnerie. En d'autres termes, si quelqu'un vous appelle, vous recevrez les caractères SEP $6C (SEP 108 en décimal) : plus besoin de détecteur de sonnerie ! Enfin, le CAP 23 est retournable en cours de connexion, ce qui n'est pas le cas de tous les Minitels (les vieux M1 qui ont fait la guerre ne possèdent pas cette possibilité).Le CAP 23 est donc un périphérique intéressant et bon marché (le prix public unitaire prévu

Vitesse & Stockage ...
Sans Limite !

MEGADRIVE

Disque Dur 44Mo/19ms Amovible Syquest™

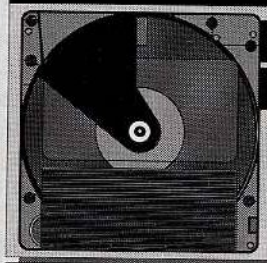
❑ MEGADRIVE 50SII
▲ 44Mo/SCSI.
▲ Amovible/CapacitéInfinie.
▲ Vitesse 19 ms (Seconde Génération).
▲ 1 Cartouche Syquest™ GRATUITE.
▲ Multiple Compatibilité : Macintosh, Atari, Amiga, IBM, ... Akai S-1100/S-1100/S-950/S-770/E-Mu ...

6990F

Disques Durs SCSI 180 & 380Mo/12ms

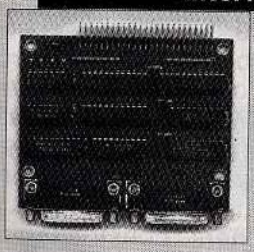
❑ MEGADRIVE 180/380
▲ Deux Versions : 180 & 380Mo.
▲ Temps d'Accès 12 ms.
▲ InterfaceSCSI.

7990F 180Mo
9990F 380Mo



Cartouches Megadrive 50SII™
Cartouches Megaflie 44™
▲ Cartouches 44Mo Garanties Origine Syquest™.
790F

Interface DMA (ACSI)-SCSI pour Atari™



990F

❑ CONVERTER
▲ Interface DMA/SCSI avec Double Port DMA/SCSI.
▲ Compatible Aladin/Spectre/PC-Ditto/Macintosh/IBM(™).
▲ Livrée avec Soft Editeur/Auto-Formatteur/Test/Auto-Boot.
▲ Fonctionne avec Plus de 100 Disques & Streamers SCSI.
▲ Edition jusqu'à 14 Partitions/Transfert 10Mb/Sec.

❑ Je souhaite recevoir une documentation .....
❑ Je souhaite vous commander .....
❑ Chèque ❑ Mandat ❑ Contre-Remboursement
❑ NOM .....
❑ ADRESSE .....
❑ CODE .....
❑ VILLE .....
TELEPHONE .....

Nos Prix sont TTC et Révisables sans Préavis Revendeurs : Nous Consulter.

DIGITAL CONCEPT INTERNATIONAL
Division Informatique

159 Rue du Faubourg Poissonnière 75009 Paris
Tel : (1) 42.82.90.89 • Fax : (1) 42.82.92.77

est de 810 F TTC), mais qui pourrait avoir du mal à trouver sa clientèle. En effet, en prenant les concepteurs à contre-pied, on pourrait dire qu'un Minitel fait tout ce que propose le CAP 23, et ce pour 20 francs par mois... L'idée reste cependant excellente, et nous attendons avec impatience le modem CAP 225 annoncé par ExtraDOS, qui devrait être aussi peu cher que le CAP 23 par rapport à ses concurrents de conception plus classique. La version de base sera conforme aux avis V21, V22, V22bis et V23 du CCITT, et une version plus évoluée intègrera les protocoles MNP 4 et V42 de correction d'erreurs, ainsi que les MNP 5 et V42bis de compression de données. Un vrai modem de luxe ! Les CAP 23 et CAP 225 devraient tous deux être disponibles à la vente courant Septembre. Donc, rendez-vous à la rentrée pour un test exhaustif de ces produits, avec photos en abondance.

CLAVIUS

Quelques petites news de chez Clavius, l'importateur français du Spectre GCR... Tout d'abord, signalons l'existence d'une version couleur (la 2.65C), qui permet d'utiliser un moniteur couleur en émulation Mac, mais il semble qu'elle doive encore connaître quelques améliorations, le codage couleur n'étant pas parfait. De plus, une telle option est très gourmande en mémoire vive, il faudra calculer ça en fonction de votre poste de travail. Dave Small travaille toujours sur la version 3.0 de GCR, dont on ne sait encore rien à ce jour, sauf qu'elle sera multilingue. Mais gageons que ce génie saura bien déclencher un certain nombre d'autres surprises... Il travaille aussi toujours sur la carte 68030 pour ST, qu'il n'est pas encore sûr de montrer à Düsseldorf à la fin Août, mais par contre, il est sûr de pouvoir y montrer la carte "MégaTalk", servant à se connecter sur le fameux AppleTalk. Rien que pour ça, on ira de toutes façons... L'autre nouveauté concernant Spectre, c'est la possibilité d'avoir, sous mode Mac, l'icône "PC Ditto" si vous disposez de cet émulateur PC. J'ai bien dit "c'est possible", ce qui signifie que Clavius attend vos courriers disant à cor et à cri que vous avez absolument besoin de cette option ! A l'aide de ces déclarations d'intention, Clavius pourra prouver à nos chers "constructeurs" qu'il y a un réel besoin...

RECTIFICATIF

Nous testions dans notre avant-dernier numéro (le 41) le logiciel musical ProScore, dédié à l'édition de partition sur ST. Mais deux inexactitudes s'étaient glissées dans notre texte : tout d'abord, il n'est pas édité par Midigam mais par "Digigram" (légère homonymie qui a provoqué notre erreur - espérons que les auteurs ne nous en veulent pas trop !). Le distributeur, quant à lui, est bien toujours la maison Comus France. D'autre part, le prix du logiciel est quelque peu supérieur à 1500 F, puisqu'il coûte très exactement 1890 F prix public TTC.

DU NOUVEAU CHEZ ICD

(à ne pas confondre avec les français "DCI"...). Ces sympathiques américains nous ont concocté une nouvelle mouture de leurs utilitaires pour disques durs absolument époustouflante. Tout d'abord, le formateur a été entièrement remanié pour simplifier encore son utilisation. Ensuite, il a été rajouté un programme de mesures de performances de disques durs, permettant de vous assurer de la bonne vitesse de transfert. Il est à noter que les résultats obtenus sont supérieurs de 25 % aux résultats annoncés par les constructeurs. Cela tient principalement à deux raisons : d'une part, l'optimisme généralement constaté chez les fabricants de disques durs quant aux performances de leur production, d'autre part à la méthode de calcul utilisée, plus proche d'une utilisation courante. Ensuite, il y a une nouvelle version (v4.8.1) du driver, la pièce maîtresse, aux performances proprement époustouflantes : elle intègre la gestion de mémoire cache de taille paramétrable, de buffers en lecture et écriture, une vérification débrayable des écritures disques, une mémoire cache paramétrable sur les FAT et les directories des disques et la coorection du bug dit "des 40 dossiers", bug qui consiste à planter méchamment et souvent irrémédiablement une partition quand on y crée plus de 40 dossiers. Il est à noter que ce bug a été corrigé dans le TOS1.4 (pour lequel il n'y a toujours pas de décision de commercialisation avancée).

Pour récupérer toutes ces nouvelles versions, deux solutions : soit appeler votre distributeur qui est tenu de vous les fournir, soit les télécharger directement sur le serveur monovoie ICD directement aux USA. Il suffit de se munir d'un modem symétrique 1200, 2400 ou même 9600 bauds et de composer le 19-1 815 968 2229. Au moyen d'un soft de télécoms de type FLASH ou INTERCOM_PC, il suffit alors de récupérer les fichiers en question. A noter que sur ce serveur, on trouve des tas d'autres fichiers ou programmes ayant rapport aux disques durs, des programmes de compression-décompression récents, et que le téléchargement est bidirectionnel : on peut bien sûr récupérer des fichiers mais on peut aussi en déposer sur le serveur si l'on pense qu'ils peuvent intéresser d'autres utilisateurs. une initiative intéressante que devraient suivre les fournisseurs européens pour la mise à jour et le suivi de leurs produits. Coût de la communication : une dizaine de francs la minute, mais ça en vaut vraiment la peine!

FORUM ATARI

La manifestation de la rentrée se confirme donc plus que jamais, et se déroulera du 18 au 21 octobre au CNIT-La Défense à Paris. 5000 m2 ont été réservés à cet effet, et l'organisation générale prévoit deux grands espaces : l'un pour le grand public et les applications ludiques, créatives et familiales en général, l'autre pour les applications pro. Mais certains exposants se retrouveront évidemment dans les deux catégories, et l'originalité des présentations tient à la mise en place "d'Ateliers thématiques en libre accès". Ce qui signifie en clair que tout sera fait pour permettre au public de "toucher", de cliquer et d'être en rapport "physiquement" avec la machine et les logiciels. Un cycle de conférences diverses sera aussi mis en place, avec des sujets généraux tels la PAO, le graphisme, la musique, etc. Sur ce dernier sujet, une conférence spécifique sera sans doute organisée avec Jean-Michel JARRE du fait de son option technique et politique sur des systèmes ST pour son travail musical. A l'occasion de ce Forum, aussi, il y aura peut-être du nouveau sur le CD-Rom (tiens, un revenant !), et le lancement du TT semble se confirmer. Mais prions, mes frères, car ce n'est pas encore officiel....

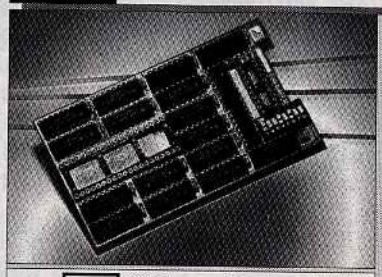
JURIMEGA 2.0

Dans notre numéro 18 (ouf, ça fait déjà un paquet de temps, ça...), nous avions déjà eu l'occasion de vous entretenir de ce logiciel dédié aux Avocats, qu'il soulage de leurs préoccupations comptables et de leurs avatars de trésorerie. En effet, ce produit, qui permet de générer une comptabilité du type bordereaux trimestriels ou Comptavocat, part de l'idée géniale qu'il n'est pas impérativement nécessaire de connaître la comptabilité pour savoir où fuit le temps et ce qu'il coûte (puisque le temps, c'est de l'argent, non ?). Le principe consiste à relier des tarifs, totalement adaptables aux besoins très personnels de chaque Avocat, avec le temps passé pour tel ou tel type de prestation, et avec le dossier individuel de chaque client pour, in fine, triturer tout pour l'exporter sous la forme d'une vraie compta... Explication : à 600 francs de l'heure, plaidoirie au palais, pour Duschpountz dans le cadre de son litige, paiement d'un huissier pour notifier l'assignation, à l'aide d'un chèque émis sur le crédit plouc se transforme en un beau crédit/ débit du 512 par le 400. Non seulement le produit gère la compta, mais il isole une véritable compta client et permet de faire des statistiques, d'éditer automatiquement des lettres de relance délicieusement judiciaires, et d'éviter que des honoraires non-perçus occasionnent de difficiles fins de mois.

Au plan de la technique informatique, le progiciel vertical a été construit autour de DBMan dernière version et reste un modèle de ce que l'on peut faire avec ce produit dont les bogues éventuels ont été contournés d'après les auteurs. La version 2.00

La Mémoire à des Prix ... Vacances !

Atari STF



Montage Gratuit en Nos Locaux.

Montage en Version : 4Mo

❑ ATAMAX 2 — 1990F
▲ Intégrable au 520, 1040 & Mega 2.
▲ Extension Mémoire Totale 2.5Mo.
▲ Extensible 4Mo par une 2è ATAMAX™.
▲ Taille Extrêmement Réduite (11cmx6 cm).
▲ Installation 5 Minutes par l'Utilisateur.
▲ Aucune Soudure ni Perçage.
▲ Circuit Multi-Couche Vernis Haute Qualité.
▲ Rams CMOS Ultra-Rapides (80-100ns).
▲ Livrée avec Notice Détaillée d'Installation.
▲ Disquette de Test Mémoire Incluse.
▲ Garantie Inconditionnelle à Vie.

❑ ATAMAX 4 — 3980F
▲ Identique à ATAMAX™ 2.
▲ Extension Mémoire Totale 4Mo.

❑ ATAMAX 0 — 1290F
▲ Identique à ATAMAX™ 2.
▲ Livrée sans Ram.
▲ Garantie Limitée.

❑ ATAMAX 1 — 680F
▲ Update 520ST > 1040ST Economique.
▲ Extension Mémoire Totale 1Mo.

Atari STE

Modules Mémoire SOJ 256Ko & 1Mo 80ns

MOTOROLA

Le Module 1Mo
590F

2Mo 1180F
4Mo 2290F

Atari est une Marque Déposée par Atari Corp.

520STE > 1040STE
2 X 256Ko 490F

❑ ATARAM 256
▲ Modules 256Ko (Extension par Paires).
▲ Haute Vitesse.
▲ Update 520STE > 1040STE Economique.
▲ Extension Mémoire Totale 1Mo.
▲ Garantie Inconditionnelle à Vie.

❑ ATARAM 1
▲ Modules 1Mo (Extension par Paires).
▲ Extensions Totales Possibles : 2.5 & 4Mo.
▲ Vitesse 70/80ns.
▲ Garantie Inconditionnelle à Vie.

• ADAPTATEUR SIM/SIP Gratuit !
• Installation Totalement Gratuite Effectuée en Nos Locaux !

❑ Je souhaite recevoir une documentation .....
❑ Je souhaite vous commander .....
❑ Chèque ❑ Mandat ❑ Contre-Remboursement
❑ NOM .....
❑ ADRESSE .....
❑ CODE .....
❑ VILLE .....
TELEPHONE .....

Nos Prix sont TTC et Révisables sans Préavis Revendeurs : Nous Consulter

DIGITAL CONCEPT INTERNATIONAL
Division Informatique

159 Rue du Faubourg Poissonnière 75009 Paris
Tel : (1) 42.82.90.89 • Fax : (1) 42.82.92.77

est bien sûr mieux écrite, améliorée en ergonomie bien que l'impossibilité de choisir ce que l'on cherche parmi une liste soit toujours là, et il est indispensable de disposer d'une machine à l'heure et à la date idoine. Les auteurs ont intelligemment trouvé le point d'équilibre entre l'apport de nouvelles fonctionnalités (comme l'édition de la DAS2) et l'amélioration des mécanismes d'origine. Avec certains logiciels médicaux bien connus, JURIMEGA se positionne dans le peloton de tête des logiciels sur ST, le seul regret est qu'il ne soit pas sous GEM, mais les juristes ont l'habitude de l'austérité et il y en a, juste là où il faut.

SALON DE LA MUSIQUE

Ce sera l'événement de la rentrée, avec des dates désormais définitives : du 12 au 18 Septembre prochain, avec 3 journées professionnelles les 16, 17 et 18. ST Mag y sera présent comme chaque année, avec le prochain "Spécial Musique", qui sortira en kiosque juste avant. Tout cela se passera comme d'habitude à la Grande Halle de la Villette (Porte de Pantin à Paris), et cette année devrait voir l'avènement de la fameuse "nouvelle formule" : nouveaux repérages, nouveaux espaces de regroupement thématiques, animations spécifiques, "nocturne" au cours des journées Pro, concerts répartis sur l'ensemble du Salon, etc. Bref, on vous y attend !...

TT : Tan'Tattendu.

Notre confrère néerlandais ST-Nieuws (à vos souhaits) annonçait il y a peu qu'Atari Benelux a présenté le TT à la presse à la fin du mois de juin. Ceci leur a permis d'annoncer deux changements importants dans les caractéristiques de cette machine, qui a tout de même déjà été montrée à Düsseldorf il y a 10 mois, et qui est encore loin d'être disponible. Le premier de ces changements a trait à la fréquence du 68030 qui est au coeur de la machine, puisqu'il sera désormais cadencé à 32 MHz au lieu des 16 longtemps cités. De quoi donner du punch à la machine ! L'autre changement se situe du côté des mémoires de masse, puisque le disque dur proposé dans la version de base (qui inclut, je vous le rappelle, outre la machine équipée de 2 Mo de RAM et de son disque dur, un moniteur couleur haute résolution), passe à 40 Mo, et profitera d'un temps d'accès moyen de 28 ms, ce qui est tout de même largement mieux que ce à quoi on avait été accoutumé de la part d'Atari. Le prix de la configuration reste toujours au même niveau, c'est-à-dire aux alentours de 25000 francs TTC. Le seul problème, comme je l'ai dit plus haut, c'est qu'on ignore encore et toujours QUAND. Voyez la "news" sur le Forum Atari à ce sujet, puisque ce serait la date la plus indiquée pour une telle sortie dans notre beau – mais souvent délaissé – pays.

T.I.M.

Ce sont les initiales de "Tactile Interactive Media", une nouvelle borne interactive construite autour d'un système ST, dont nous avons vu un exemplaire tout récemment. Développée par la société Mosaïc ("les bâtisseurs d'images interactives") basée à Sophia Antipolis dans l'arrière-pays niçois, et distribuée par la société FutureVision (186 rue du Faubourg St Martin, Paris 10ème), cette borne présente plusieurs originalités. Tout d'abord, elle gère une unité centrale ST, un disque dur Megafile 30, un Genlock SATV pour les incrustations vidéo, et surtout... un vidéodisque, en l'occurence de marque Philips. Toute cette configuration est évidemment aménageable en fonction des besoins du client (mémoire, type de vidéodisque, etc.), mais l'atout majeur du système est d'être livré avec un générateur d'applications, développé sur ST, qui permet au client -ou au prestataire de service- de programmer lui-même l'applicatif de sa borne, en gérant son arborescence, le type de séquences (vidéo, images fixes, images ordinateur), la langue (français, anglais, allemand, italien) et la présentation générale de l'écran, notamment le mode "standby" (veille et "réveil"). TIM dispose d'un écran tactile très performant, puisque ce dernier permet d'isoler les surfaces sensibles pratiquement au pixel près (sur écran basse résolution soit 320x200), et nous avons pu découvrir un applicatif complet très bien présenté, relatif à la base d'informations de Sophia Antipolis (congrès, manifestations, etc.). Présenté sous forme de "Livre d'Or", avec des pages dessinées à l'écran et comportant un coin retourné, le plus spectaculaire était évidemment de "tourner les pages" avec un doigt, en respectant le mouvement habituel que l'on a pour tourner les pages d'un vrai livre. Remarquable, cet effet était accompagné d'une gestion de la gravité et de l'inertie lorsqu'on "rate" son mouvement... et que la page "retombe" ! Cette structure "Livre d'Or" est fournie avec le générateur d'applications, qui est lui-même limpide. Présenté en couleur, il se présente sous la forme de choix successifs en cliquant dans les cases appropriées, et le script généré reste modifiable à tout moment. Le driver de vidéodisque fonctionne parfaitement, et permet d'aller spécifier les images ou les séquences d'images en précisant leurs numéros. Au point de vue sécurité (en cas de micro-coupures, ou de plantage inopiné), TIM dispose d'une séquence de lancement automatique pour assurer la reprise sans intervention humaine. Extérieurement, la borne dispose d'un design intéressant, et il est certain que ce type de développement va connaître une grande diffusion dans les années à venir, ne serait-ce que dans les lieux publics. En tous cas, ça marche avec un ST et avec de la vidéo qúant au prix, il est évidemment fonction de la configuration choisie.

BIONET

Le fameux réseau de BioData, importé en France par ALM, ne cesse d'évoluer et désormais, il présente une interface X-Windows pouvant tourner sur ST. Auparavant, le ST intégré dans ce réseau pouvait toujours accéder au monde Unix en matière de transferts divers, grâce au mode VT52, mais maintenant il pourra même, à condition de disposer de 2 mégas de mémoire vive, se transformer en terminal X-Windows ! D'autre part, à la demande des utilisateurs, le réseau Bionet va pouvoir fonctionner avec un serveur ST uniquement, et "alimentera" un réseau limité à 3 ou 4 postes. Par contre, ce serveur pourra fonctionner en tâche de fond. Cette version-là sera terminée pour la fin Juillet, et sinon, dans les "grands desseins", l'objectif "Mac" est plus que jamais à l'ordre du jour, avec peut-être une présentation à l'Atari Show de Düsseldorf à la fin du mois d'Août prochain...

OFFREZ-VOUS LA QUALITÉ GOLDSTAR® AU PRIX DU BULK !

Disquette 3'' 1/2
720 K SFDD ~~9,90 F TTC~~
(Réf. MF1D)

5,90 F TTC *(par 100)*

Egalement sur catalogue :
Disquette 5'' 1/4 - 1,2 Mo *(Réf. M2HD)*

CONTROL RESET A PARIS

▼ **PARIS 8** : 38, rue de Turin - Tél. (1) 45 22 51 00
▼ **PARIS 12** : 60, cours de Vincennes - Tél. (1) 43 40 80 80
▼ **PARIS 15** : 44, rue de Cronstadt - Tél. (1) 48 42 55 10
ET 22 AUTRES BOUTIQUES EN FRANCE

Control reset

CLAVIUS et l'émulation parfaite

Gadgets by Small, Inc, Colorado USA

SPECTRE GCR

Ecrit et lit Mac directement sans transfert !

- Tourne avec ROMS 128 K, sur STF, STE et STacy – L'écran est 30 % plus large
- Copie d'un fichier de 500 K sur disque dur passe de 2mn 26 à 8 secondes chrono
- Le Floppy (A ou B) écrit 8 fois plus vite – Rafraichissement écran 400 % plus rapide
- Vitesse globale du micro émulé 20 % plus rapide qu'un Mac.
- Laser SLM 804 sous Postcript/Ultrascript, Quickdraw. Matricielles reconnues
- Version 2.65 c dispo. Nous renvoyer votre disquette originale

Avec ROMS 128 K : **N.C.**

La carte MegaTalk (Apple Talk + port SCSI pour Mega ST 1, 2, 4) devrait atteindre les côtes de France vers la fin août 90.

DVT-Scope : Back up de disque dur sur toute entrée/sortie vidéo de magnétoscope VHS ou autre. 360 M sur une T 120 1290 F+ port 30 F

ST 4096C

La palette des 4096 couleurs du 520 STE dans tout ST et Mega ST ! **690 F**

MEGA•TOUCH - *Mettez du ressort dans votre ST !...*
95 ressorts à placer (facilement !) sous les touches de votre clavier pour une sensation Mega ST : **140 F** + port 20 F

KEY-MARKS
Réglettes s'installant devant les touches de fonction du ST dans lesquelles, grace à un logiciel d'impression fourni, vous afficherez leurs significations selon l'application utilisée : **160 F** + port 20 F

Avant-Garde Systems, Floride USA

pc-ditto II

Un produit hardware meilleur qu'un XT Turbo à 10 Mhz

Indice Norton SI de plus de 4.0. Accès à une librairie de plus de 15000 logiciels même protégés. PC Ditto II est le seul émulateur qui vous donne vraiment l'EGA, EMS, MIDI, les liaisons série jusqu'à 19200 bauds et l'impression Laser sur SLM 804.

De hautes performances ne peuvent être obtenues que si le port DMA de votre ST reste libre de toute contrainte en dehors de son rôle propre. Installation par simple "clipage" dans votre ST (manuel en français). Pas de boitier annexe qui encombre votre bureau. Emulation souris série MicroSoft (tm).

Complètement transparent : continuez d'utiliser GCR, GEM, et autre.

Il suffira d'une simple mise à jour de la disquette programme pour redéployer sa force et ses nouvelles possibilités.

Offrez-vous la puissance au plus juste prix.

2 690F Installé

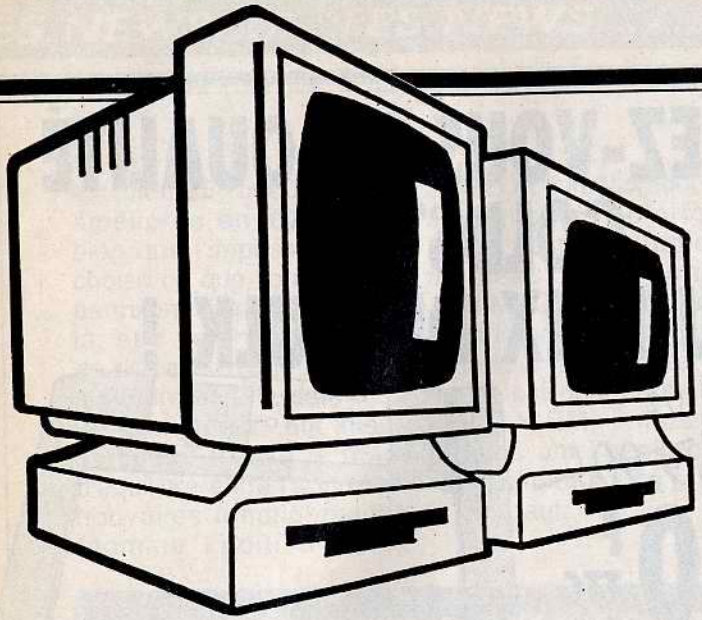
G.E. Box I - Boitier métallique (à l'abri des rayonnements) pour tout disque dur à la norme SCSI comprenant carte interface DMA/SCSI, alimentation, roue codeuse, câbles de raccordements et soft de gestion disque dur (jusqu'à 255 partitions et plus de 100 M°!)
Prêt à l'emploi : **1 990 F** + port 90 F

IMG Scan : le célèbre scanner fibre optique se fixant sur tête d'imprimante – Nouveau prix – **990 F** complet + port 30 F.

CLAVIUS *sera fermé du 15 Août au 2 Septembre*

Tous ces produits sont disponibles chez votre revendeur ou chez **CLAVIUS**

19, rue Houdon, 75018 PARIS - Métro Pigalle
☎ 42.62.90.19 — Fax : 42.62. 95.85



SOURIS : DU NOUVEAU

La souris fournie avec les ST ne restera pas dans les annales de la micro comme un record de qualité, de fiabilité, d'esthétisme et d'adéquation aux environnements souvent difficiles que peuvent rencontrer ces petits animaux. Voici que d'un seul coup, trois concurrentes tentent de s'imposer, avec chacune des avantages (ou inconvénients) différents.

LOGIMOUSE

La Logimouse de Logitech est probablement la souris la plus "design" du lot (je ne vais pas vous la décrire, regardez plutôt la photo). Sa forme de tranche de demi-tube (NDLR : non, non, il ne la décrit pas) surprend assez au début, et la position des boutons vous interdit de laisser votre main affalée dessus, comme vous le faites avec la souris du ST. Mais une fois que vous avez pris l'habitude de la tenir du bout des doigts, elle se révèle très largement supérieure au bout de plastique gris qui vous sert actuellement de souris: elle est légère, les boutons sont larges et montés sur des ressorts très agréables, et le câble de liaison est nettement plus long que celui de la souris ST standard.

Cette souris est de plus livrée avec un accessoire de bureau, qui vous permet de régler la vitesse de déplacement du pointeur, si jamais vous trouvez qu'il se traîne vraiment. Cet accessoire n'est pas le meilleur accélérateur de souris qu'il m'ait été donné de voir, car il n'accélère pas le pointeur proportionnellement au déplacement de la souris, mais il est tout de même de bonne qualité.

Si jamais vous en avez assez de votre souris, si vous voulez la remplacer par un périphérique un peu plus beau et un peu plus agréable à utiliser, et si vous n'êtes pas prêt à investir dans la Golden Image, la Logimouse est probablement le meilleur choix.

CONTRIVER MOUSE

Cette souris étant la moins chère des trois, elle convient parfaitement comme solution de remplacement en cas de perte, panne ou vol de votre souris ST. Dans ces trois cas, vous aurez en effet plutôt tendance à essayer de minimiser le plus possible les frais de remplacement, et n'irez pas chercher les souris les plus luxueuses du marché...

La Contriver Mouse est d'un esthétisme assez douteux, à moins que vous n'appréciez les tons de beige brunâtres, et les boutons, non contents d'être assez bizarrement placés, sont extrêmement durs (encore plus que ceux de votre souris habituelle). Mais sa forme est agréable et rappelle vaguement celle de la souris ST, bien qu'elle soit un peu plus petite, alors que la Logimouse et la Golden Image sont radicalement différentes.

Elle est livrée avec un support vous permettant de l'accrocher sur le côté de votre moniteur (l'adhésif manquait à l'appel, mais c'est un détail, c'est pour ça que je le met entre parenthèses), ainsi qu'avec un joli tapis de souris assorti à la teinte inoubliable de votre ST. Enfin, contrairement à la Logimouse, la Contriver n'est pas fournie avec une démo du jeu Pipe Mania, ce qui méritait d'être précisé.

GOLDEN IMAGE

Telle est la marque de la troisième souris dont nous débattrons ce jour, et qui a l'illustre honneur d'être importée et distribuée par Incoma. Cet animal de la classe des périphériques (perifericus ordinarus), et de l'espèce de ceux d'entrée (intra data), appartient à la race des souris dites optiques (opticus sorix), c'est d'ailleurs ce point qui la distingue des autres souris.

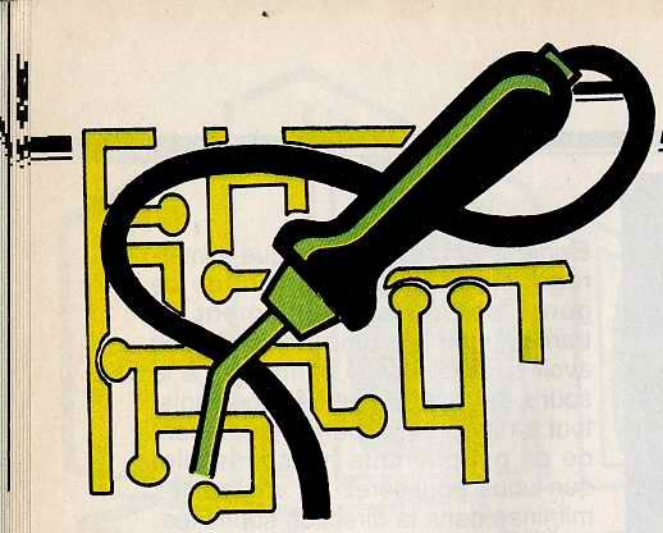
"Optique" n'est pas ici applicable à la transmission entre la souris elle-même et l'ordinateur, comme c'est le cas de la BMC, que nous avons testée il y a quelques mois. "Optique" est donc ici lié à la méthode de repérage des déplacements de cet objet si utile en ces temps voués aux interfaces graphiques, qui se fait à l'aide de ce qui ressemble à deux LEDs et une loupe sur la partie inférieure de la souris, et d'une trame portée par le tapis de souris fourni.

Le tapis en question est, lui aussi, totalement différent des autres ustensiles portant le même nom et ayant le même usage (je vous le rappelle, accueillir la souris, et en faciliter le déplacement). En fait, on trouve la classique surface de mousse, réduite à sa plus simple épaisseur (il y a quand même deux millimètres), dont le but est d'assurer l'adhérence vis-à-vis de la surface sur laquelle est sauvagement jeté ce tapis. Mais, au dessus de cette mousse, on trouve une surface plastico-cartonnée (ou cartonno-plastiquée ? Ça fait un peu terroriste...) qui porte, comme je vous l'ai dit, une jolie trame formée de points blancs sur fond gris, l'ensemble ayant finalement une couleur proche de celle des ST.

Evidemment, on ne peut que s'interroger sur le fonctionnement de ce genre de souris. Visiblement, la trame qui orne le tapis de souris doit avoir quelque usage permettant à la souris de repérer ses déplacements, tout en assurant le parfait glissement de ce périphérique indispensable, que vous pousserez de vos petites mimines dans la direction souhaitée. Petites mimines, soit dit en passant, qui pousseront les boutons qui en forment la proue. Ils sont au nombre de trois, ce qui est pour le moins inhabituel sur ST, mais seuls deux d'entre eux sont utilisés, celui du milieu étant lâchement abandonné car destiné aux PC. Ils réagissent au quart de poil, ce qui fait très plaisir après les souffrances endurées pendant de longues années avec la souris d'origine.

Cette souris a le gros avantage d'être inusable mécaniquement, puisqu'il n'y a pas de partie mécanique pour les déplacements, et que sa toilette se fait en deux coups de chiffon, pas besoin de se livrer à un démontage en règle. Comme en plus elle n'est pas laide, qu'on a un tapis de souris (qui lui est indispensable, c'est vrai), et que les boutons sont très bien, on ne peut que lui reprocher son prix relativement élevé (plus de 500 F), mais pour du "haut de gamme"; comme le dit son distributeur, pourquoi pas ?

Watsit & Jacques Caron



LE ML68705 DEV-P1

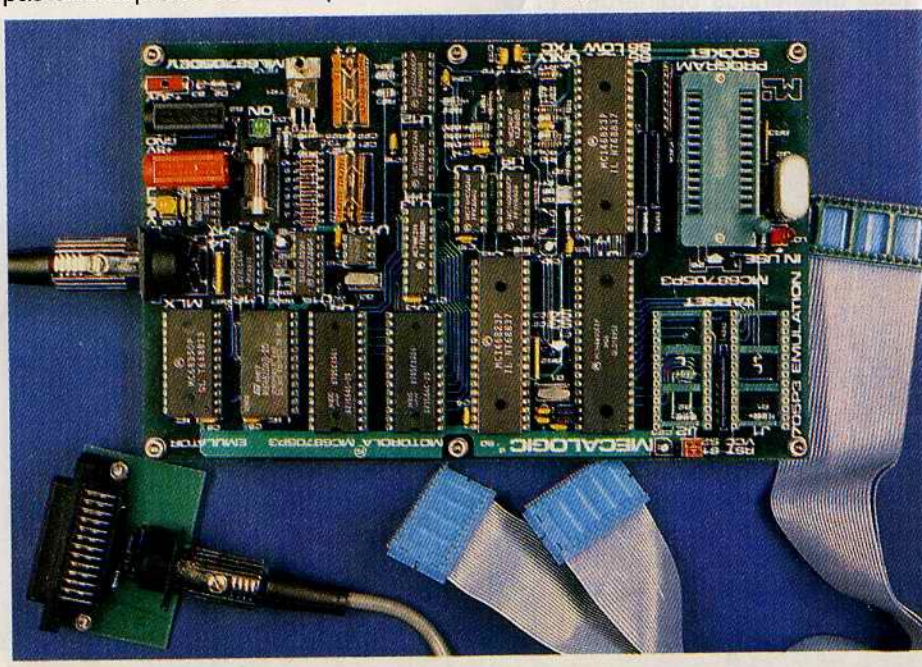
Non, ce n'est pas la référence d'un composant dernier cri de la famille 68000, mais tout simplement le nom de code d'une carte de développement pour le microcontrôleur 68705P3. Il s'agit d'une carte qui se branche sur votre machine et qui permet de développer et de mettre au point du code pour le 68705, un microcontrôleur bien connu des électroniciens de tout acabit. Mais voyons avant tout ce qu'est un microcontrôleur...

QU'EST-CE DONC ?

C'est une petite bête de 28 broches qui contient un microprocesseur (1, 4, 8, 16 ou même 32 bits pour les plus performants), quelques octets de RAM, quelques kilo-octets d'EPROM, des ports d'entrées/ sorties pour communiquer avec l'extérieur et des timers programmables. Le 68705, lui, est un microcontrôleur 8 bits et dispose de 112 octets de RAM utilisateur, de 1804 octets d'EPROM, de 20 bits d'entrées/ sorties et d'un timer programmable. Vous allez me rétorquer qu'avec si peu de mémoire, on ne doit pas pouvoir faire grand-chose. Eh bien détrompez-vous! Une foule d'applications se contente de microcontrôleurs car on voit, d'après ce qui précède, que ce composant rassemble, certes en petite quantité, tous les constituants d'un ordinateur, et possède son propre langage d'assemblage lui permettant de piloter toutes sortes de petites applications : décodage de clavier, gestion de lignes série, programmateurs d'appareils électroménagers, commande de moteur pas à pas, alarmes, décodeurs-désembrouilleurs, bref toute une panoplie d'applications relativement simples et répétitives ne nécessitant pas une capacité de calcul phénoménale. L'opération est financièrement intéressante car un microcontrôleur coûte infiniment moins cher que son équivalent en circuits classiques de capacité équivalente (microprocesseur + RAM + EPROM + circuits d'interface). Il remplace à lui seul une bonne dizaine de circuits différents, réduisant du même coup la complexité du circuit imprimé de la carte ou du montage développé. De plus, étant programmable et même reprogrammable, cela permet des modifications ou des mises à jour plus faciles et rapides qu'avec des circuits logiques classiques, de type TTL par exemple.

LE KIT DE DEVELOPPEMENT

La carte proprement dite se connecte d'un côté sur le port série du ST et de l'autre, par l'intermédiaire d'un câble en nappe, à l'application à développer. Par application, il faut



comprendre ensemble hard+soft destiné à remplir une fonction quelconque. Il est à noter que ce kit, utilisant la liaison série, marche aussi bien sur ST que sur PC. Il suffit de demander la version de logiciel adaptée à la machine, rien sur la carte elle-même ne différençiant les machines connectées.

La carte électronique (200x125 mm) est très propre : circuit double face, trous métallisés, pas de fils ou de straps rajoutés à la dernière minute, tous les composants sont sur supports, un fusible est là pour protéger la carte. En résumé, une carte qui paraît bien faite même si on ne peut s'empêcher de penser qu'elle pourrait être un peu plus compacte.

MISE EN OEUVRE

La carte à mettre au point sera branchée au kit de développement durant toute la phase d'écriture et de mise au point du code, jusqu'à la programmation du microcontrôleur. Un câble en nappe ramène tous les signaux importants de la carte vers le kit pour simuler totalement et fidèlement le 68705. Il faut bien saisir que l'application sera totalement dépendante de l'ordinateur-hôte.

Ceux qui ont SPECTRE (128 ou GCR), et qui ont lu les "News-letters" jusqu'au bout, auront vu que c'est ce genre d'outil, bien plus puissant que n'importe quel débogueur soft, qu'utilisait Dave Small pour mettre au point son émulateur MacIntosh (cf. Gadgets News Letter 2, ZAX "In Circuit Emulator"), c'est dire que c'est loin d'être un gadget et que le marché porte sur des dizaines de millions de dollars !

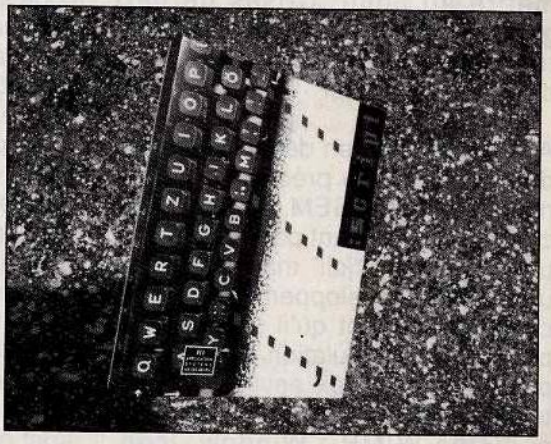
A QUOI CA SERT ?

Le kit intervient à toutes étapes du développement de sa carte : écriture du code source, assemblage, génération de code objet, chargement de ce dernier dans la carte, mise au point et programmation finale du microcontrôleur, une fois le code finalisé. Son usage permet pratiquement de décupler la vitesse de mise au point.

En effet, contrairement au déboguage de programmes en assembleur

QUE

pensez-vous d'un traitement de textes dont la prise en main serait immédiate, et qui comblerait tous vos désirs d'édition ?
En quelques «clics» de souris, définissez vos blocs de texte instantanément, soulignez-les, mettez-les en forme, copiez-les, enrichissez vos documents. En haute résolution bien sûr, en parfait WYSIWYG évidemment, et surtout... avec les centaines de polices au format Signum!, qui vous procureront des sorties de la plus haute qualité, y compris sur les matricielles. Quant bien même la vitesse d'impression serait pour vous cruciale, vous pouvez aussi exploiter les polices, proportionnelles ou non, de votre imprimante.
Mais simplicité et performance ne sont pas tout. Vous disposez d'une palette de fonctionnalités des plus étendues : insertion de graphiques, compatibilité ASCII, macros-fonctions, raccourcis clavier, tabulations décimales, notes de bas de page, etc...



o prouvez-moi que performance et simplicité sont conciliables : ci-joint un chèque de 50 F, pour une version bridée de :script.
o je souhaite simplement recevoir une documentation.
o :script coûte-t-il réellement 790 F seulement ?
o :script tourne sur Atari ST 1 Mo haute résolution. Que puis-je faire avec mon moniteur couleur ?

NOM: _____ PRENOM: _____
N°: _____ RUE: _____
CP: _____ VILLE: _____
Coupon à renvoyer à : **Application Systems Paris**
18, rue Germain Dardan 92120 Montrouge
Tél.: (1) 40 92 80 81

1
2
3
4

:script

Le traitement de textes définitif.

sur micro-ordinateur, la programmation et la mise au point sur microcontrôleur se font un peu "en aveugle" : entendez par là que, faute de ressources suffisantes (mémoire RAM, ROM, dispositif de visualisation, accès aux registres, etc.), on ne voit pas bien ce qui peut se passer. De plus, sachant que l'application tourne au minimum à 1 MHz (soit un million de cycles d'horloge, chaque instruction prenant entre un et quatre cycles), n'espérez pas pouvoir saisir des phénomènes aussi fugitifs. Sans kit de développement, on n'a ni accès aux registres, ni possibilité de mettre des points d'arrêt. Sans kit de développement c'est comme si, sur votre machine, vous preniez un assembleur quelconque, que vous écriviez votre programme source, que vous l'assembliez et que vous essayiez, sans l'aide d'un débogueur et sans possibilité de traçage, de le mettre au point. Crises de nerf garanties et mise au point d'applications complexes reportée aux calendes grecques!

Avant d'aborder en détail l'utilisation du kit, une petite précision : le soft n'est pas sous GEM, le souci des concepteurs n'ayant pas été de faire un programme "joli" mais de fournir un outil de développement riche et complet, d'autant qu'il s'adresse, a priori, à une population de développeurs habitués aux environnements textuels (par opposition aux environnements WIMP : Window, Icon, Mouse, Pointer). A partir de maintenant, on "range" donc sa souris... On lance le programme qui intègre toutes les fonctions nécessaires par la suite. Un bandeau rappelle toutes les commandes disponibles à un moment donné. Ces commandes sont activables par l'appui sur la touche correspondant à la première lettre du nom de l'option (A pour Assembler, E pour Editer...) et la touche ESCAPE permet de sortir d'une option.

L'EDITEUR-ASSEMBLEUR

C'est un "éditeur page" avec toutes les fonctions classiques: insertion, suppression, notion de bloc, etc. Au moyen des touches de curseur, on peut se déplacer n'importe où dans le texte, l'écran affichant une page de 24 lignes. L'assembleur est du modèle dit "symbolique à deux passes" : dans un premier passage, il récupère tous les symboles, c'est-à-dire tous les noms de variables et les étiquettes des branches du programme. Au cours de sa deuxième passe, il génère le code proprement dit. Il est assez tatillon sur la syntaxe à respecter. Chaque ligne se compose de champs optionnels mais séparés par au moins un espace : champ étiquette, champ instruction, champ opérande, commentaire. Le tout est facile à prendre en main et le cycle édition-assemblage est très rapide.

LE DEBOGUEUR

On attaque maintenant la partie la plus intéressante : l'émulation sur le kit de développement de l'application, et le déboguage et traçage du code objet. Par l'intermédiaire d'une commande, on charge le code objet, non pas dans sa carte d'application mais dans la carte d'émulation elle-même, ce qui permet d'avoir accès à un pseudo-68705 idyllique : on a accès à tous ses registres, on peut mettre des points d'arrêt un peu partout, faire du pas à pas, examiner et modifier totalité ou partie de la mémoire, modifier les registres et repartir, et bien d'autres choses encore. Toutes choses utiles pour mettre au point rapidement et efficacement son application et qui ne sont pas possibles en temps normal.

Généralement arrivé à ce point là, et sauf don ou chance exceptionnels, il y a de grandes chances pour que vous repartiez dans l'éditeur-assembleur car il paraît peu probable que votre programme marche du premier coup. On repart donc dans un certain nombre de cycles édition - assemblage - deboguage jusqu'à ce que le programme tourne correctement.

Le code étant finalement au point, il ne reste plus qu'à programmer le 68705 définitif qui, le malheureux, se morfondait dans son coin. Pour ce faire, rien de plus simple, il suffit de le mettre à l'emplacement prévu sur la carte et d'utiliser la commande de programmation. Le code étant maintenant inscrit dans le microcontrôleur, il ne reste plus qu'à le mettre en place sur la carte développée et à couper le cordon ombilical qui la reliait à la carte de développement. A partir de ce moment seulement, votre montage est réellement autonome.

LA DOCUMENTATION

Elle est en français, volumineuse (250 pages) et remarquablement didactique, expliquant en détail le fonctionnement du microcontrôleur et reprenant toutes ses instructions en détail. On trouve aussi une bonne vingtaine d'exemples dont le source se trouve sur la disquette, et permettant de se faire rapidement la main. On y trouve également un long chapitre sur le fonctionnement de l'émulateur lui-même, sur ses connexions, sur ses limites et différences par rapport au microcontrôleur émulé.

CONCLUSION

J'espère que cette première excursion dans le domaine des microcontrôleurs vous aura intéressé et donné l'envie d'aller y voir de plus près. C'est un monde aussi passionnant que celui des micro-ordinateurs même si ses applications, de par les ressources mémoire limitées, ne sont pas du tout comparables. Le plaisir de voir tourner un montage électronique autonome est aussi grand que celui de voir tourner un programme. La carte, proposée à moins de 5000 F par la société aixoise MECALOGIC, n'est certes pas donnée pour un budget d'amateur, mais est incomparablement moins chère que les autres solutions professionnelles qui existent. Elle a sa place dans tout bureau d'études et dans les organismes de formation en électronique : écoles, centres de formation professionnelle. De plus, tournant indifféremment sur ST et sur PC, elle dispose assurément de la base installée la plus large qui soit. Que pourrait-on demander de plus ? Peut-être une nouvelle version du soft plus conviviale tournant sous GEM et la mise en boîtier de la carte...

DiskMaster

KOMELEC KOMELEC KOME

4, rue Yves TOUDIC – 75010 PARIS

Du lundi au samedi de 10 h à 12h h 30 et de 13 h 30 à 19 h – métro République

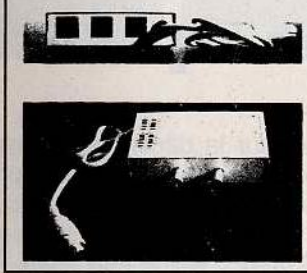
Tél.: 42.08.63.10 – 42.08.54.07 Fax : 42.08.59.05

LA CONNECTIQUE A DES PRIX EXCEPTIONNELS

EXTRAIT DE NOTRE CATALOGUE CONNECTIQUE

Lecteur ATARI Switch

300 F



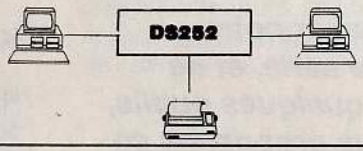
Commutateur Video

200F

commutateur vidéo Mono/Couleur

Sortie Audio (type RCA)pour Ampli.Ext.

DS252	DS254	DX252
210 F	310 F	351 F



CABLE ROND AU METRE

14 Non Blindé	17.00
14Blindé	26.00
cable en nappe au métre 0.40 le PTS	
14 Conducteur	5.60

SOURIS 3 BOUTONS
COMPATIBLE
ATARI STF / STE
200 F

41256 – 10	35 F
41256 – 12	32 F
43256 – 8	145 F
43256 – 10	135 F
27256	30 F
27C256	32 F
27512	86 F

CABLE INFORMATIQUE SUR ATARI ET AMIGA

Cable peritel male 13 Broches male 150.00	
quadrupleur de joystick ATARI ST 1 DB25 / 2DB9	150.00
doubleur de joystick ATARI ST 1 15 hd / 2DB9	150.00
Cable minitel PC / ATARI ST/ AMIGA DB25F / DIN 5 Br	95.00
Cable péritel amiga/ DB23M/ 2 RCA M / 1 peritel male	180.00
Cable péritel amiga/ DB23F/ 2 RCA M / 1 peritel male	180.00
Cable disque dur ATARI DB19M / DB19M .70m	120.00
Cable pour connecter leteur 5"1/4	130.00
Cable pour connecter leteur 3"1/2	130.00
Rallonge peritel M/F ATARI ST / AMIGA	135.00
Rallonge peritel M/M ATARI ST / AMIGA	55.00
Rallonge peritel M/M Croisé	55.00r
Rallonge joystick 1.80m	70.00
Rallonge joystick 0.30m	40.00
Rallonge Moniteur DIN 13 males/ 13 Femmelles 2m	130.00
Rallonge Moniteur DIN 14 males/ 14 Femelles 2m	100.00
Prise ATARI ST 13M Br (moniteur)	20.00
Prise ATARI ST 13 F Ci(lecteur)	20.00
Prise ATARI ST 13 F(cordon)	30.00
Prise ATARI ST 14M Br	20.00
Prise ATARI ST 14F Ci	20.00
Prise ATARI ST 14F Cordon	30.00
Prise DB9 Male ou Femelle	8.00
Prise DB23 Male ou Femelle	8.00
Prise D25 Male ou Femelle	4.00
Capot DB19 Vis longues	8.50
Capot DB23 Vis longues	8.50
Capot DB25 Vis longues	6.80
Free Boot	100.00
Autres longeurs: supplément au métre linéaire Prix TTC	

PROMO KOMELEC SUR LES CABLES

CABLE IMP/PC	PAR 1	PAR 10	+ 10
1.80 m	50.00 F	40.00 F	N.C.
3.00 m	95.00 F	78.50 F	N.C.
5.00 m	129.00 F	110.00 F	N.C.
7.00 m	170.00 F	144.50 F	N.C.
CABLE SERIE			
1.80 m	59.50 F	38.00 F	N.C.
3.00 m	96.00 F	79.50 F	N.C.
5.00 m	135.00 F	115.00 F	N.C.
7.00 m	175.00 F	147.15 F	N.C.
10.00 m	222.00 F	185.00 F	N.C.
SUB D	**PAR 1**	**PAR 50**	**PAR 100**
DB09 M/F	4.05 F	3.25 F	2.71 F
DB15 M/F	5.00 F	4.57 F	4.03 F
DB19 M/F	8.00 F	7.84 F	7.06 F
DB23 M/F	8.00 F	7.84 F	7.06 F
DB25 M/F	4.50 F	4.10 F	3.35 F
DB37 M/F	10.00 F	10.60 F	9.52 F
DB50 M/F	25.00 F	22.51 F	20.26 F

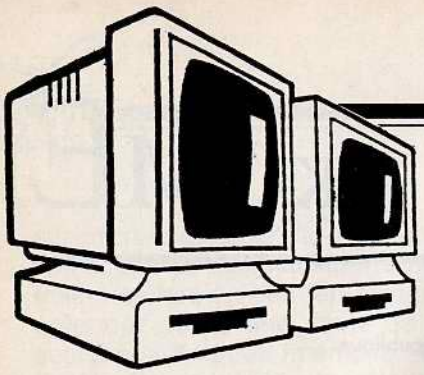
ATELIER DE CABLAGE

A votre disposition
conception de tous type de câble

Vente par Correspondance : Mini 100 F. Catalogue 30 F remboursé à partir de 200 F d'achat.

Port 40 F = MAXI 3 KG. Contre Remboursement avec 20 % à la Commande.

Administrations Acceptées. Tout nos prix sont donnés T.T.C. Prix indicatifs. Prix par quantité nous consulter.



LES EXTENSIONS MÉMOIRE

4e épisode

Afin d'égayer notre habituelle série, et de combler quelques oublis, nous vous proposons ce mois-ci une réalisation destinée aux mordus des cocottes en papier. En effet, nous vous offrons la possibilité de mettre à jour votre vieux numéro 42, et ce, grâce à de subtils collages. Vous trouverez tout au long de cet article les instructions nécessaires au bon déroulement de la manipulation, qui est à la portée d'un enfant de moins de 10 ans (même notre assistant télématique l'a réussie, alors pensez-vous...).

MODE D'EMPLOI

N'ayez aucune crainte, tout se passera bien !

- Munissez-vous de ciseaux et d'un peu de colle.
- Découpez proprement les colonnes suivantes, en prenant soin de ne pas les mélanger.
- Repérez la page 50 du précédent numéro.
- Collez avec soin les textes découpés, en lieu et place de ceux du mois dernier.
- Et rachetez ce numéro, de manière à conserver un ST Mag 43 en parfait état.

Malgré le soin que nous avons porté à cette réalisation, il se peut que vous rencontriez de légers problèmes. Dans ce cas, veuillez nous contacter, en mentionnant sur votre lettre "Service consommateurs ST Magazine, Rubrique pliages et collages". En outre, si vous possédez un Minitel, nous vous rappelons que notre serveur 3615 STMAG est ouvert nuit et jour, et même le dimanche.

PROTECHNIC

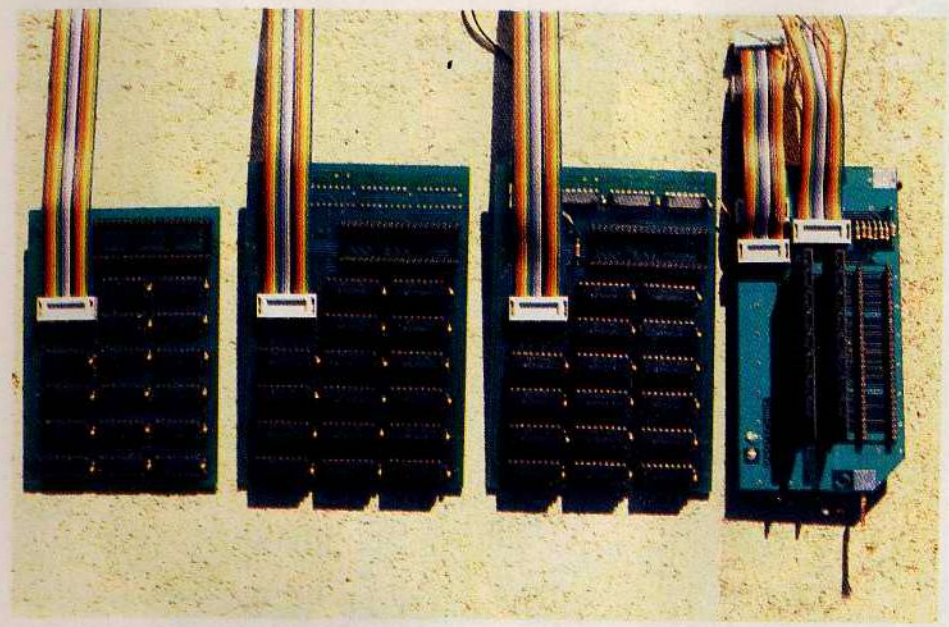
Une information incomplète est la cause de toutes les manipulations que vous venez de faire subir à votre mensuel. En effet, notre précédent article sur Protechnic ne présentait pas toute la gamme de ses produits, et introduisait alors des erreurs. Pour exemple, nous annoncions qu'une partie de la mémoire des 1040 était perdue lors du passage en 2,5 mégas; mais c'était sans compter sur la déclinaison 3 mégas de l'extension, spécialement prévue pour cette machine.

LA GAMME

La toute première carte de cette société date des 520 ST avec lecteu de disquette et alimentation séparés Bien qu'elle ne soit plus disponible aujourd'hui, on peut souligner qu'elle s'est vendue à plus de 400 exem plaires, nombre important pour le parc de machines de l'époque.

Puis vint le modèle pour 520 STF dont un millier de machines sont à ce jour équipées (sans compter les copies marquées Protechnic !). On peut se demander pourquoi une telle carte a vu le jour, alors que le STF offrait directement la possibilité d'une extension de 512 Ko. La réponse est simple : il s'agissait de diminuer le nombre des soudure (17 contre 256, qu'il fallait préalable ment dessouder !). Elle est toujour proposée, et à un prix légèremen supérieur à 600 F.

Le troisième produit existe en deu versions, prévues pour les modèle STF. La première étend un 520 2.5 Mo, et la seconde un 1040 à Mo. Contrairement à ce qui éta annoncé le mois dernier, le még interne du 1040 est donc bien util sé, puisque la carte n'apporte qu deux mégas supplémentaires. Pa contre, cette récupération de l RAM d'origine complique son insta lation, et a pour conséquence d'e empêcher la vente en kit (contraire ment à l'extension pour 520).



La toute dernière carte, nommée Delta, fonctionne avec des barrettes SIPP (des SIMM avec des pattes). Elle peut être installée dans les Méga 1 et 2, ainsi que dans les STF. De plus, prévue pour ne pas encombrer le Méga ST, elle laisse de la place libre aux PC Speed et autres Hypercaches...

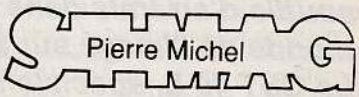
Les deux cartes 2 Mo et la Delta équipée de deux barrettes 1 méga sont disponibles à un prix public d'environ 2500 F.

Ces produits Protechnic étant exclusivement disponibles auprès des revendeurs, nous vous conseillons d'étudier les prix publics pratiqués.

Certaines marges peuvent en effet augmenter sensiblement les prix, et l'on trouve parfois les 2 Mo aux environs de 3000 F !

RÉCAPITULATIF

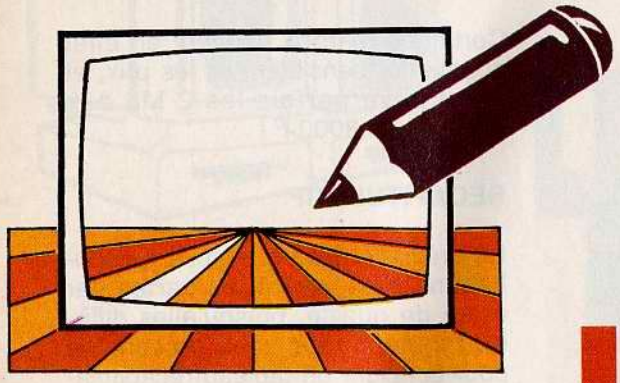
En précisant bien que nous ne souhaitons pas classer les cartes par ordre de qualité, puisqu'elles diffèrent toutes, nous vous présentons pour conclure un tableau récapitulatif des offres proposées.
Vous noterez toutefois qu'il n'est pas exhaustif, et tous les distributeurs concernés par la question de l'extension mémoire sont invités à contacter la rédaction afin que nous présentions leurs solutions et aboutir ainsi à une présentation complète de l'offre en ce domaine.



Fabricant	Nom	Config.	Total	Kit	Soudures	Installation	Prix approx.
Clavius	DIL	520STF	1 Mo	Non	Oui	Comprise	1000
Clavius	DIL	STF	2,5 Mo	Non	Oui	Comprise	2700
Clavius	DIL	STF	4 Mo	Non	Oui	Comprise	4000
Clavius	DIL	M2	4 Mo	Non	Oui	Comprise	2500
D.C.I.	Atamax 0	STF, M2		Oui	Non	Gratuite	Nue 1300
D.C.I.	Atamax 1	520STF	1 Mo	Oui	Non	Gratuite	700
D.C.I.	Atamax 2	STF	2,5 Mo	Oui	Non	Gratuite	2000
D.C.I.	Atamax 2	M2	4 Mo	Oui	Non	Gratuite	2000
D.C.I.	Atamax 4	STF	4 Mo	Oui	Non	Gratuite	4000
Frontier Soft.	Xtra-RAM	520STF	1 Mo	Oui	Non	300 F	1200
Frontier Soft.	Xtra-RAM	STF, M1	2,5 Mo	Oui	Non	300 F	2500
Micro-Vidéo		M1	2 Mo	Non	Oui	Comprise	2000
Micro-Vidéo		M1	4 Mo	Non	Oui	Comprise	3500
Micro-Vidéo		1040STF	2 Mo	Non	Oui	Comprise	1500
Micro-Vidéo		STF		Non	Oui	Comprise	4 Mo SIMM 3500
Protechnic	Méga STF	520STF	1 Mo	Oui	Oui	Comprise	700
Protechnic	2 Méga STF	520STF	2,5 Mo	Oui	Oui	Comprise	2500
Protechnic	2 Méga STF	1040STF	3 Mo	Non	Oui	Comprise	2500
Protechnic	Delta	STF, M1, M2		Oui	Oui	Comprise	2 Mo SIP 2500
Sud Micro		STF	2,5 Mo	Non	Oui	Comprise	2200
Sud Micro		STF	4 Mo	Non	Oui	Comprise	3800

- M1 et M2 signifient Méga ST 1 et Méga ST 2.
- La colonne Total donne la mémoire après extension.

Dernière minute : Vous trouverez dans le tableau ci-dessus deux produits de la société Sud Micro, dont nous avons reçu les caractéristiques au moment du bouclage...



LE GST 40 E

Voilà qui va apporter un souffle d'air frais dans le monde de l'image sur ST ! Le GST 40 E, spécialement conçu pour tout type de STe, est en effet le premier genlock-incrustateur à fonctionner réellement sur votre machine préférée, qu'on se le dise ! C'est une société française (cocoricooo...), Satellite & Télévision, qui est à l'origine de sa création, en s'attachant à offrir, avec une extrême facilité d'emploi, un résultat d'une excellente qualité.

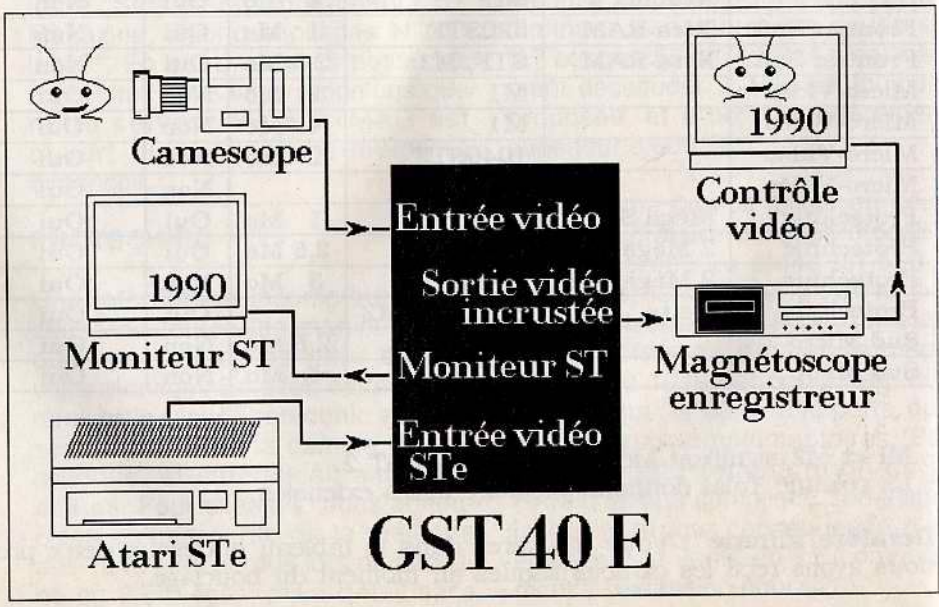
LES JOIES DE LA VIDEO

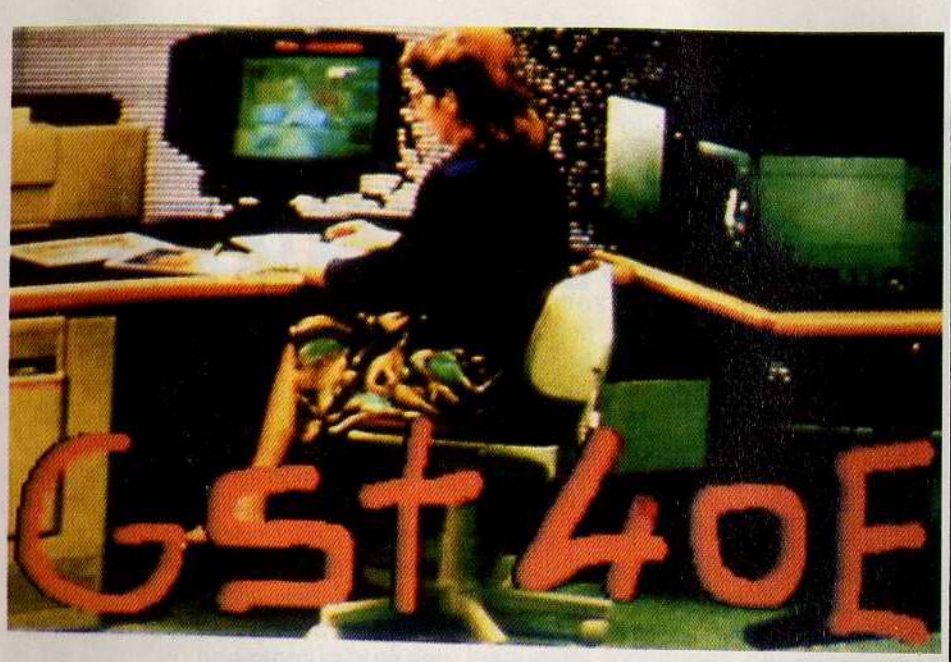
Incruster proprement une image en provenance du STe sur une image vidéo est donc désormais possible. Il est important de rappeler succintement le rôle des trois parties principales qui composent un tel boîtier. Tout d'abord, le genlock synchronise les deux sources vidéo pour avoir les deux images (STe et vidéo) parfaitement l'une sur l'autre. Il s'agit ensuite d'encoder en PAL les composantes Rouge, Verte et Bleue du STe, avant d'incruster l'image du STe sur l'image vidéo. La couleur de fond de l'image STe est considérée comme transparente pour la vidéo, et l'on voit ainsi apparaître l'image vidéo en arrière plan à travers cette découpe.

L'installation du GST 40E ne devrait vous poser aucun problème. En vous aidant du schéma ci-joint, vous arriverez très rapidement à enregistrer sur bande vidéo la date de la dernière visite de votre ami le martien. Les connections vidéo s'effectuent sur le devant du genlock, celles du STe à l'arrière. Le boîtier que vous pouvez voir sur la photo est en fait un modèle de présérie, la prise audio sur la face avant ne figure plus dans la version finale commercialisée.

COMMENT ÇA MARCHE

Procédons par ordre : la sortie moniteur du STe se connecte au GST 40E (câble fourni), puis en ressort





vers le moniteur du STe. La caméra se branche sur l'entrée vidéo du GST, et enfin, il ne vous reste plus qu'à relier sur la sortie vidéo un quelconque magnétoscope PAL (VHS, 8mm, U-matic...).

Une fois la configuration en place, booter le STe avec la disquette fournie, le logiciel "Desync. Prg" installé dans un dossier AUTO signifiant au STe qu'il doit se synchroniser sur une horloge externe. Il suffira de régler ensuite la phase sur la face avant du genlock, de façon à ce que l'image soit d'une qualité optimale.

Et hop ! L'installation ci-dessus vous permet d'obtenir, sans aucun problème, une incrustation en moins de 5mn. Simplicité garantie ! Voilà de quoi laisser rêveur les genlocks plus perfectionnés et "boutonneux", qui aiment généralement se laisser tripatouiller pendant un certain temps (un temps certain!), avant de daigner afficher un premier résultat...

Le signal vidéo (camescope) ne fait que passer à travers le GST 40E sans subir aucune modification. On le retrouve en sortie avec la même qualité, décoré des zibouibouis du STe incrustés par dessus. Petit détail pratique, si vous débranchez ladite entrée vidéo, vous coupez ainsi l'arrivée de l'horloge qui synchronise l'ensemble. Le GST 40E recrée alors automatiquement une synchro, pour éviter tout plantage du STe. En fait, ce cas de figure correspond à une rupture de signal vidéo, réel à chaque fois que vous arrêtez la lecture du camescope, ou que vous rembobinez. Le GST 40E, grâce à cette particularité, peut également être utilisé comme encodeur PAL. Il suffit de ne pas connecter l'entrée vidéo, seule l'image du STe sera disponible en sortie. Vous pourrez ainsi copier vos images et animations STe sur votre magnétoscope.

Une version spéciale du GST 40, est aussi disponible pour les ST. Par contre, il faudra faire preuve d'un peu plus de patience, et d'un minimum de dons en bricolage pour en profiter. En effet, seul le STe a été prévu pour accepter directement un genlock. Les propriétaires de ST devront donc ouvrir leur machine pour placer un petit sabot sur l'un des circuits intégrés.

Son prix inférieur à 2300 F et sa simplicité de mise en oeuvre doivent permettre à un plus grand nombre de vidéastes de profiter d'ouvertures graphiques performantes, réservées jusqu'à présent aux initiés, ou à ceux qui avaient plus de moyens. Une version Y-C de ce genlock, dédiée uniquement aux magnétoscopes S-VHS, est en cours de réalisation.

Anne Olivelli

ArtSoft

Les Sources ArtSoft

Le concept qui a fait notre succès : des programmes exécutables performants et originaux, développés pour répondre aux besoins fréquemment rencontrés par les utilisateurs, et accompagnés de leur fichier source (listing) largement commenté.

Six disquettes disponibles (250 F. chaque) :
1: Comptes Bancaires + Utilitaires
2: Edition Etiquettes + Compactage + Divers
3: Editeur Disque + Simulation + Utilitaires
4: Environnement de travail sur ST
5: Gestion de Base de données
6: Utilitaires Anti-Virus (Détection, Vaccin, ...)

Les Utilitaires ArtSoft

Voici, regroupés sur une même disquette, 15 Utilitaires et Accessoires indispensables à votre confort et celui de votre ST.

Quelques-unes des fonctions couvertes :
HEURE : mise à l'heure automatique après reset, affichage permanent, réveil.
ECRAN : capture de l'écran, harcopy sur imprimante, auto-extinction de l'écran.
MEMOIRE DE MASSE :ram-disque, recherche de fichiers, lecture/impression de fichiers ASCII ou Hexadécimaux, vaccination des disquettes.
SYSTEME : Reset au clavier, taille mémoire disponible, accélérateur de souris intelligent, détection des virus.

La disquette est livrée avec une notice d'utilisation imprimée ..... (Prix 390 F.).

Jeux de Lettres ArtSoft

(pour Mots Croisés, Mot le Plus Long, SCRABBLE, Anagrammes, ...)

Pour les amateurs de jeux de lettres, ce programme contient un dictionnaire évolutif de plus de 60.000 mots, et dispose de fonctions de traitement et de recherche très évoluées.
Il dispose de filtres multiples pour le tri et la recherche : lettres en vrac ou pré positionnés, genre des mots et forme, longueur des mots, etc...

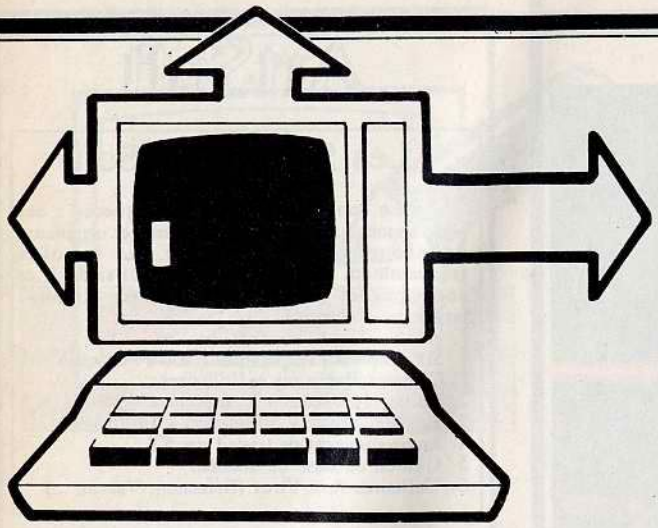
Fonctionnant avec toute la gamme ST, du 520 ST au Mega ST, en monochrome ou couleur, ce programme est livré avec notice d'utilisation imprimée ..... (Prix 390 F.).

Disquette de Présentation
et de Démonstration.
(avec Source) ..... 60 F.

☐ Disquette / Programme .....
☐ Disquette de Démonstration ..... 60 F.
Participation aux frais de port ..... 15 F.
☐ Liste détaillée (joindre enveloppe timbrée)

Ci-joint mon réglement par chèque à :

DUTAUD
11, rue François Gillet
42000 Saint-Etienne



PUBLIC ASTRO

VERSION 2

Du nouveau pour les fanas d'astrologie, puisque les éditions MEZANGUEL proposent la toute dernière version de PUBLIC ASTRO (voir ST Mag n• 25). Ce logiciel déja très vaste par ses différentes applications a subi une refonte très importante. La documentation a été encore plus clarifiée qu'auparavant surtout pour les premiers pas avec le logiciel (préparation d'une disquette de travail qui servira pour toute la recherche astrologique) ainsi que quelques pages supplémentaires concernant la protection du logiciel. Que le lecteur et surtout l'auteur, veuillent bien nous excuser de la mauvaise qualité des illustrations qui accompagnent cet article, la conversion des images couleur en noir et blanc provoquant immanquablement une perte de définition et des déformations inesthétiques. Ces images ne doivent préjuger en aucun cas de la présentation réelle du logiciel en moyenne résolution, des problèmes techniques nous ayant empêché d'insérer les images couleur adéquates.

Public Astro vous propose désormais 2 possibilités pour les saisies :
- soit une saisie par le mode questions/réponses ;
- soit une saisie par mode graphique à l'aide d'un tableau par l'intermédiaire de la souris. Ce mode est très pratique en utilisation, car il n'y a aucune action sur le clavier, et il est très simple de corriger une erreur puisque la sélection d'une nouvelle valeur remplace le choix précédent.

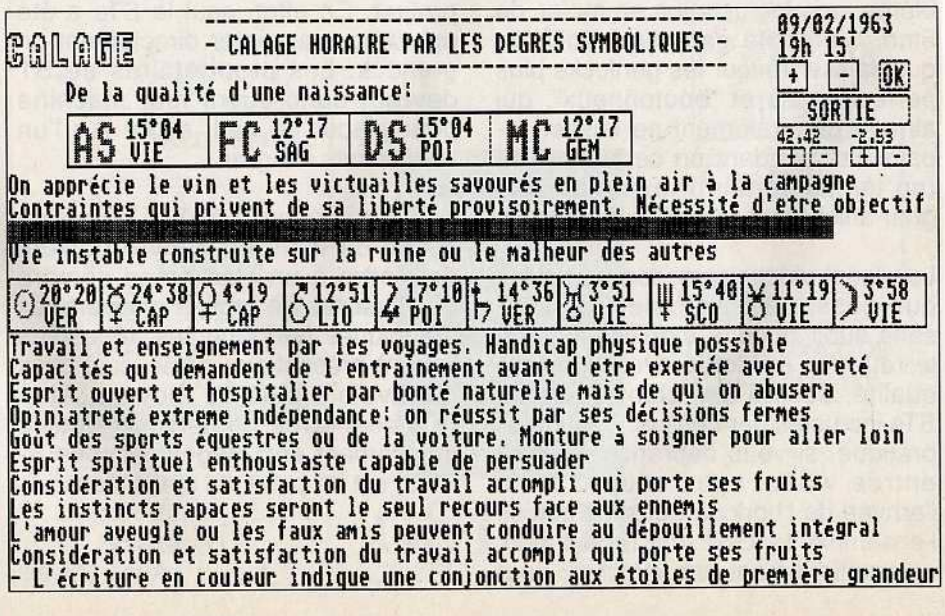
Le calage horaire permettant de faire varier l'heure de naissance à la minute près reste inchangé (voir fig. 1).

En ce qui concerne la domification, vous pouvez sélectionner trois méthodes différentes : CAMPANUS, REGIONMONTANUS, ou PLACIDUS. L'option thème pose la question sur l'âge concerné et le lieu d'anniversaire.

Ecrit en GfA 3.5, donc plus rapide que l'ancienne version, le thème est dressé en 30 secondes, et l'ensemble des études comprend :
1) L'horoscope natal ;
2) Les directions symboliques pour l'âge demandé ;
3) Les âges aux transits par les directions symboliques ;
4) L'horoscope progressé ;
5) Thème de révolution solaire ;
6) Thème de révolution lunaire.

Les parties 1, 4, et 5 comportent chaque fois : une grille d'éphémérides (fig. 2), une carte du ciel correspondant (fig. 3), et un tableau de consultation. Chacun des choix peut être bien évidemment imprimé.

Pour un astrologue professionnel, ceci va représenter la partie principale, car elle va lui permettre de



gagner un temps considérable. Il est évident qu'il utilisera ses propres interprétations par rapport à tout son savoir d'astrologue. Le novice a la possibilité d'utiliser la disquette 'interprète' (dont la majorité des interprétations ont été tirées du "Dictionnaire astrologique" de H.-J. GOUCHON) qui est encore plus fournie, et devra en dégager une synthèse, ce qui représente pour l'utilisateur de nombreuses heures d'analyse personnelle vu la quantité d'interprétations.

Un "plus" également avant tout calcul, c'est le tableau de correction qui permet la modification des longitudes planétaires, écarts en minutes d'angles. Vient ensuite le calcul des directions primaires. Le logiciel permet de commencer par l'étude des directions avec leur interprétation, puis d'utiliser les transits pour y ajuster une date. Les fichiers créés peuvent être consultés à tout moment par l'option lecture, et sont utilisables pour toute une vie, et à chaque aspect sur toute une vie correspond une interprétation au jour près.

Les transits, quant à eux, concernent toutes les planètes sauf la Lune et Pluton. Lors de la lecture des transits, le logiciel trace deux graphiques, l'un schématisant sous la forme de courbes les déplacements planétaires, l'autre représentant la puissance des transits planétaires en rapport avec la position des éléments du ciel natal. L'évaluation se fait par la force d'aspect et de position chiffrée et proportionnée aux bonnes et mauvaises influences.

Le calcul des variations angulaires entre chaque planète demeure inchangé. L'option "Composite" permet de sélectionner 2 fiches pour lesquelles longitudes des planètes et maisons seront calculées. La carte du ciel est alors dressée et doit être interprétée à vue. L'application définit le type de relations entre deux individus et leur exprime leur devenir potentiel. On peut vérifier également entre deux événements, personne-événement, personne par rapport à une date, etc., puisqu'il s'agit de fusion entre deux thèmes.

La grosse nouveauté (et de taille !) étant l'option localisation, qui apporte une disquette supplémentaire (ce qui en fait 4), et prend en considération l'heure et la date comme valeur fixe, le lieu géographique pouvant être choisi à loisir sur une carte mondiale (les valeurs sont tirées de l'Atlas de Poche de la Librairie Larousse). Par exemple, il suffit de choisir un lieu de naissance à New York, et la carte sera orientée sur les Etats-Unis. Il suffit ensuite de cliquer à un endroit choisi pour voir apparaître longitude et latitude. Un quadrillage peut se superposer à l'écran si on le désire pour faciliter la recherche des coordonnées. Ensuite, par simple appui d'une touche, la carte du ciel est dressée pour ce lieu. Pratique, non ?

Révélez la puissance

PAINT DESIGNER

Un outil de dessin en mode BITMAP et VECTORIEL pour ATARI monochrome. Reconnait de nombreux formats d'images, dont le format utilisé par les logiciels de PAO.

PAINT DESIGNER allie puissance et simplicité grâce sa nouvelle interface utilisateur.

Un logiciel à la mesure du talent de votre ordinateur

ATARI avec 1 méga de mémoire minimum – Monochrome uniquement

Prix conseillé: 590 Francs 2 disquettes + manuel

NOUVEAUTE

BASIC DESIGNER

Il met votre disposition plus de 70 commandes trés puissantes pour réaliser des démonstrations ou des Slideshows d'images, avec l'ouverture vers de nombreux formats reconnus.

Pour ceux qui désirent aller encore plus vite, le module ANIMATOR réalise SANS PROGRAMMATION, l'animation voulue.

BASIC DESIGNER posséde un véritable compilateur qui transforme votre programme BASIC en un programme exécutable sous GEM.

ATARI avec 1 Méga de mémoire minimum – Monochrome uniquement

Prix conseillé: 450 Francs 1 disquette + manuel

NOUVEAUTE

PAYS DU MONDE

Vous apprendrez avec PAYS DU MONDE à situer en vous amusant les 170 pays souverains de la planète. Si vous hésitez, des aides spécifiques pour chacun de ces quelques 220 pays et territoires vous mettront sur la voie de la solution. Vous apprendrez également à classer les pays selon leur population, leur superficie, ou leur produit national brut par habitant.

ATARI 520,1040 et Méga – Couleur et monochrome

NOUVEAUTE

BON DE COMMANDE

Je commande exemplaire (s) du logiciel au prix de

+ 25 FF de frais de port

Réglement bancaire – Carte bleue – ou Mandat

ESAT SOFTWARE 55 rue du Tondu 33000 Bordeaux Tél. : 56.96.35.23.

LE NUMERO 1 DU LOGICIEL UTILITAIRE

SOLEIL NOIR : 25° 3' BEL | EXTRAIT D' EPHEMERIDES | LUNE NOIRE : 25° 3' BAL
-(PRIAPE)- = PODOPHAG = -(LILITH)-
DOMIFICATION PLACIDUS

* EPHEMERIDES POUR LE THEME NATAL : T.S.= 4h43'10''

- Latitude : 42° 42' Nord.
- Longitude : 2° 53' Est.

- DATE : Le SAMEDI 9/ 2/1963 A 19h15

- M1 : 15°04' VIE - M2 : 9°17' BAL
- M3 : 8°39' SCO - M4 : 12°17' SAG
- M5 : 16°40' CAP - M6 : 18°07' VER
- M7 : 15°04' POI - M8 : 9°17' BEL
- M9 : 8°39' TAU - M10: 12°17' GEM
- M11: 16°40' CAN - M12: 18°07' LIO
- ⊕ : 1°25' POI

= EPHEMERIDES =

☉ SOL	☽ LUN	☿ MER	♀ VEN	♂ MAR	♃ JUP	♄ SAT	♅ URA	♆ NEP	♇ PLU	N.AS
20°20' ♒ VER	3°58' ♍ VIE	24°38' ♑ CAP	4°19' ♑ CAP	12°51' ♌ LIO	17°10' ♓ POI	14°36' ♒ VER	3°51' ♍ VIE	15°40' ♏ SCO	11°19' ♍ VIE	29°34' ♋ CAN
14°42' Sud	12°50' Nord	20°07' Sud	20°41' Sud	21°18' Nord	6°02' Sud	17°13' Sud	10°50' Nord	14°49' Sud	20°04' Nord	29°34' ♑ CAP
STATIONS	<---	--->	--->	<---	--->	--->	<---	--->	<---	N.DES

* NOTA : SOLEIL A 20°19'33" EN VERSEAU - OBL. ECLIPTIQUE: 23°26'39" -

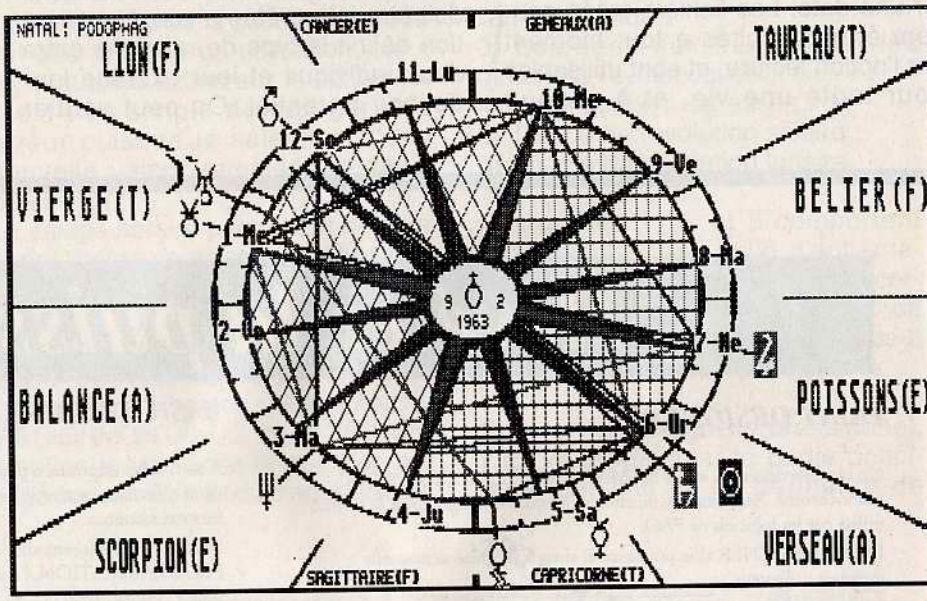
Les applications de l'option localisation sont multiples : pour un individu, la recherche d'un lieu géographique peut s'appliquer au thème de révolution solaire. Si l'individu souhaite orienter les effets vers une expansion professionnelle, il cherchera les lieux géographiques où Jupiter et Maison 10 ont la même longitude zodiacale dans le thème de révolution solaire. En astrologie mondiale, cela permet l'étude des éclipses solaires.

Le débutant en astrologie aura de quoi s'occuper pendant de très nombreuses heures rien que pour l'étude de son thème et pour saisir toutes les possibilités et les subtilités de ce logiciel. Public Astro gagne en rapidité par rapport à son prédécesseur pour le calcul et le tracé des courbes (environ 5 fois plus rapide), il peut être également installé sur disque dur.

L'auteur a réalisé un travail très soigné autant sur le soft que sur la notice. Une protection par code a été mise en place juste avant le chargement du logiciel mais qui n'est pas du tout astreignante. L'auteur envisage cependant de décomposer le logiciel en différents modules pour permettre au départ un accès encore plus simple au grand public, car le logiciel nécessite pour l'instant un minimum de 1 Mo de mémoire, et un écran couleur.

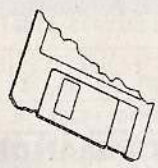
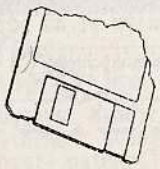
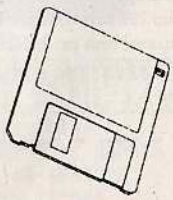
Public Astro est vraiment une référence sur le ST en tant que logiciel d'astrologie. A posséder absolument pour toute personne intéressée par le sujet, car on y découvre l'astrologie, on en comprend les mécanismes, pour enfin en être passionné à ne plus en dormir...

Michel Centelles



Quel soft préférez-vous obtenir?

Celui-ci? Celui-là? Ou celui-ci?

3615 TN2 3615 BUBSJ 3615 STMAG

LE TELECHARGEMENT

LES PACKS CADEAUX amie

PACK N° 1 :
10 LOGICIELS, 1 SUPER MANETTE

PACK N° 2 :
PACK N° 1 + 1 TAPIS SOURIS,
10 DISQUES VIERGES,
100 LOGICIELS DU DOMAINE PUBLIC.

amie LE PRO.

PACK CADEAU MÉGAPAGE
Logiciel LE RÉDACTEUR
+ Logiciel TIME WORK PUBLISHER.

ATARI 520 STE & 1040 STE

ATARI 520 STE
+ PACK CADEAU No 1 ou 50 DISQUETTES 3" 1/2
3 490 F

ATARI 520 STE
+ MONITEUR COULEUR
+ PACK CADEAU N° 2 ou 100 DISQUETTES 3" 1/2
5 490 F

-5 %

ATARI 1040 STE
+ PACK CADEAU N° 2 ou 70 DISQUETTES 3" 1/2
4 490 F

ATARI 1040 STE
+ MONITEUR COULEUR
+ PACK CADEAU N° 2 ou 120 DISQUETTES 3" 1/2
6 490 F

DU 1er JUILLET AU 30 AOUT 1990

SUPER PRIX D'ÉTÉ

-4% à -15%
sur prix affichés

UNIQUEMENT POUR REGLEMENT COMPTANT (chèques ou espèces)

REMISES NON CUMULABLES DANS LA LIMITE DES STOCKS DISPONIBLES

ATARI MEGA ST

MEGA ST4
+ 100 LOGICIELS DU DOMAINE PUBLIC
13 500 F

MEGA ST4
+ MONIT. MONO SM 124
+ Pack cadeau MEGAPAGE
14 700 F

-6 %

PORTFOLIO
~~2.990~~ F **1.990 F**

PÉRIPHÉRIQUES

-7 %

LECTEURS
3" 1/2 Ext. PROMO 790 F
5" 1/4 Externe 1 650 F

DISQUES DURS
MEGA FILE 30 3.990 F
MEGA FILE 60 6.990 F
MEGA FILE 44 8.000 F
CARTOUCHE MEGA FILE 44 990 F

MONITEURS
SM 124 1.200 F
COULEUR SC 1425 2.490 F
MULTISYNCHRO 4.990 F

TABLETTES GRAPHIQUES
PRINT CRP A4 PROMO 2.490 F
PRINT CRP A3 8.490 F

SCANNERS
PRINT TECHNIC 4.990 F
HANDY SCANNER 2.990 F

SON
ST REPLAY 4 670 F
ST REPLAY PRO 1.300 F
MASTER SOUND 540 F
FM MELODY MAKER 790 F

VIDÉO
PRO 89 2.290 F
VIDI ST + ZZ DIXIMAGE 2.300 F

ÉMULATEURS
PC SPEED 2.490 F
SUPER CHARGER 2.890 F
SPECTRE GCR 3.890 F
PC EDITO II 2.690 F

DIVERS
SOURIS ANCO 390 F
TRACK BALL 390 F
TUNER TECH 1.190 F
SOURIS BMC 895 F
COMMUTATEUR N/B COULEUR 200 F
COMMUTATEUR LECTEUR 300 F

IMPRIMANTES

-8 %

CITIZEN
120 D 1.590 F
SWIFT 24 3.990 F

STAR
LC 10 1.890 F
LC 10 COULEUR 2.300 F
LC 24-10 3.190 F

EPSON
LX 800-400 2.400 F
LQ 500-400 3.790 F

COMMODORE
MPS 1230 1.590 F
MPS 1500 COUL. 2.300 F

DISQUETTES 3" 1/2 DF DD
par 100 : **5,00 F** l'unité
par 50 : **5,50 F** l'unité
par 10 : **6,00 F** l'unité

BOÎTES DE RANGEMENT (avec clés)
50 DISKS 50 F
90 DISKS 90 F
POSSO 139 F

LOGICIELS

EXTRAIT DE NOTRE CATALOGUE

-10 %

ALL TIME FAVORITE 200 F
CASTLE MASTER 220 F
COLORADO 220 F
CONQUEROR 220 F
CYBERBALL 170 F
DEFENDER OF THE EARTH 190 F
DRAGON BREATH 270 F
EXTASE 210 F
GOLD OF AMERICA 180 F
HARRICANA 230 F
HEAVY METAL 180 F
ITALY 90 200 F
KICK OFF II 170 F
KLAX 170 F
LES AVENTURIERS 220 F
MAUPITI ISLAND 240 F
MIDWINTER 230 F
PROJECTYLE 240 F
ULTIMA V 260 F
WARHEAD 240 F
WORLD CUP YEAR 90 240 F

LES Plus d'amie

COMMANDEZ 43.57.48.20

- GARANTIE 2 ans
- CRÉDIT 4 mensualités sans intérêt*
- REPRISE Votre vieil ordinateur repris à 50 % de sa valeur**
- REMISES aux collectivités et comités d'entreprise.

* Après acceptation du dossier
** Pour tout achat d'une unité centrale de plus de 5 000 F

TÉLÉCOPIE : 47.00.50.51

VPC	11, bd Voltaire 75011 Paris	**43.57.48.20**
ATARI	11, bd Voltaire 75011 Paris	**43.57.96.89**
AMIGA	11, bd Voltaire 75011 Paris	**43.57.96.18**
PC	19, bd Voltaire 75011 Paris	**43.38.18.09**
SERVICE TECHNIQUE	13, passage du Jeu-de-Boule 75011 PARIS	**43.57.82.05**
OCCASION		**43.38.46.40**
MARSEILLE LOISIRS	69, cours Lieutaud 13006	**(16) 91.42.50.42**
MARSEILLE PC	69, cours Lieutaud 13006	**(16) 91.47.74.11**

A RETOURNER A : AMIE VPC 11, BD VOLTAIRE 75011 PARIS
NOM _____
ADRESSE _____
VILLE _____
CODE POSTAL _____ TÉL _____
MON ORDINATEUR _____

(Tous nos prix sont TTC, les promotions ne sont pas cumulables)

DESIGNATION	QUANT.	PRIX	MONTANT
FRAIS D'ENVOI*			

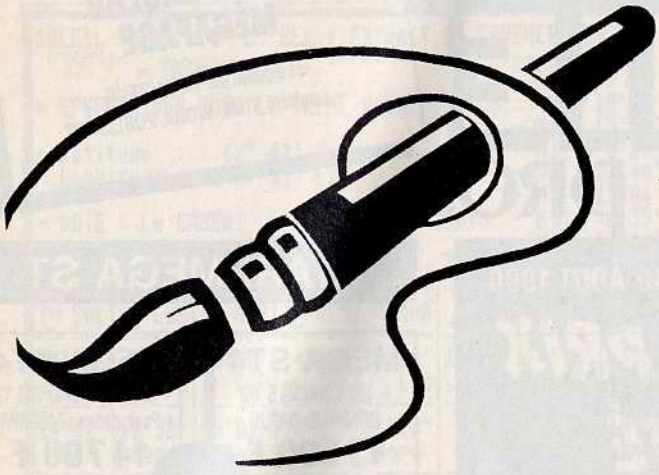
POSTE 30 F/TRANSPORTEUR 90 F C.R. 60 F TOTAL

☐ CHEQUE ☐ CCP ☐ CARTE BLEUE ☐ CARTE CLUB AMIE

DATE D'EXPIRATION

DATE _____ SIGNATURE

SCOR - STM 8.90 - RC 86 B 2526



ARABESQUE

Après EASY-DRAW 3.0 le mois dernier, nous continuons notre présentation des logiciels de dessin vectoriel, et accueillons aujourd'hui dans nos colonnes ARABESQUE. Annoncé à deux reprises dans les news, ce produit n'a cessé de nous faire saliver, et nous attendions avec impatience sa sortie dans l'Hexagone.

Comme prévu, les éditions Upgrade se sont entièrement occupées de la traduction, tant du manuel que du logiciel. A ce sujet, la préversion de la documentation qui nous a été fournie laisse présager un produit fini de très bonne qualité. Quant au programme, il était dans sa version définitive, et fort bien francisé.

Ces fleurs étant lancées, nous pouvons entamer la description de ce soft allemand, en débutant par son installation. Il fonctionne sur tous les modèles de ST, à l'exception des 520. Toutefois, les utilisateurs de laser SLM804 ne pourront tirer parti de leur imprimante s'ils ne possèdent pas un minimum de deux

Une image en mode bitmap

Déformations, tramés, remplissage de formes...

mégas (comme d'habitude). Au chapitre des moniteurs, seuls les monochromes sont utilisables, et bientôt les grands écrans. Enfin, l'utilisation de disque dur est possible, et ce logiciel peut "tourner" sous le switcher TWIST II (voir quelque part dans ce numéro).

GÉNÉRALITÉS

A demi-GEM, Arabesque utilise les fenêtres, mais laisse de côté les menus déroulants (sauf dans un mode précis qui permet d'ouvrir les accessoires, en "basculant" sur un écran spécifique). Toute la philosophie du logiciel repose sur des menus pop-up, qui apparaissent à la position de la souris après un clic sur le bouton droit. Un peu déroutant au premier abord, on s'habitue très vite à cet environnement.
Ces menus contiennent une multitude d'icônes, relativement explicites, et toutes doublées d'un raccourci clavier (quel gain de temps). Leur utilisation est très simple, puisqu'un clic gauche exécute la fonction désirée, alors qu'un droit permet d'en modifier les paramètres. Prenons comme exemple le dessin à main levé, avec pour icône un crayon. Le bouton de droite laisse donc apparaître une fenêtre de réglage de ce crayon, avec l'épaisseur du trait, son style, la forme des extrémités, etc. Puis le bouton de gauche fait disparaître le menu, et donne accès à la page de dessin. Il est possible d'afficher sur celle-ci une règle, graduée en centimètres, en pouces, en points ou encore dans votre propre échelle. De plus, les coordonnées de la souris sont constamment visualisées, ainsi que divers renseignements sur l'opération en cours (largeur, hauteur, angle...).

MODE BIT-MAP

Comme le laisse présager ce titre, Arabesque n'est pas exclusivement vectoriel, puisqu'une importante partie bit-map le compose. C'est même son mode par défaut, expliquant ainsi la position de ce paragraphe dans l'article.

Avec un maximum de 9989 par 9999 points, ce logiciel sait théoriquement gérer une résolution de 1200 points par pouce, ou une page de 84 cm de côté en 300ppp! Cependant, il faudrait lui fournir une mémoire de plus de 12 mégas, ce qu'un ST ne peut faire à ce jour. En vous limitant donc aux performances d'un Méga 4, vous pouvez travailler sur un peu plus de deux pages A4, avec une précision de 300 ppp.

Hormis les habituels outils (dessin à main levée, lignes, rectangles, cercles...), de nombreuses fonctions sont présentes. Ainsi, des courbes, des arcs d'ellipse, des triangles et des parallélogrammes 2D et 3D vous viennent en aide. De plus, un puissant mode texte permet d'écrire dans les quatre directions, en gras, léger, italique, souligné et contour. Les polices utilisées sont de type GEM classique, mais le point fort réside dans l'absence de GDOS, permettant de charger une fonte en cours d'utilisation. Signalons enfin la présence d'un utilitaire de conversion de polices Signum, élargissant grandement la bibliothèque de caractères utilisables.

Le remplissage de surface offre lui aussi des possibilités inhabituelles, telles que l'emploi de trames d'impressions, ou un mode de dégradé automatique ! Mais la partie la plus impressionnante du mode bitmap est la manipulation quasi temps-réel des blocs. En effet, après une découpe, pas moins de 24 fonctions vous sont proposées. Elles vont des plus simples, comme les miroirs et l'inversion, aux plus compliquées, avec le lissage et le remplissage de formes ! Citons aussi le détourage, les rotations, les courbures et autres déformations, les changements de taille, et les ombrages. Imaginez que pour coller un bloc à une sphère, il suffit de dessiner un cercle, et de remplir ce cercle avec le bloc... Impressionnant !

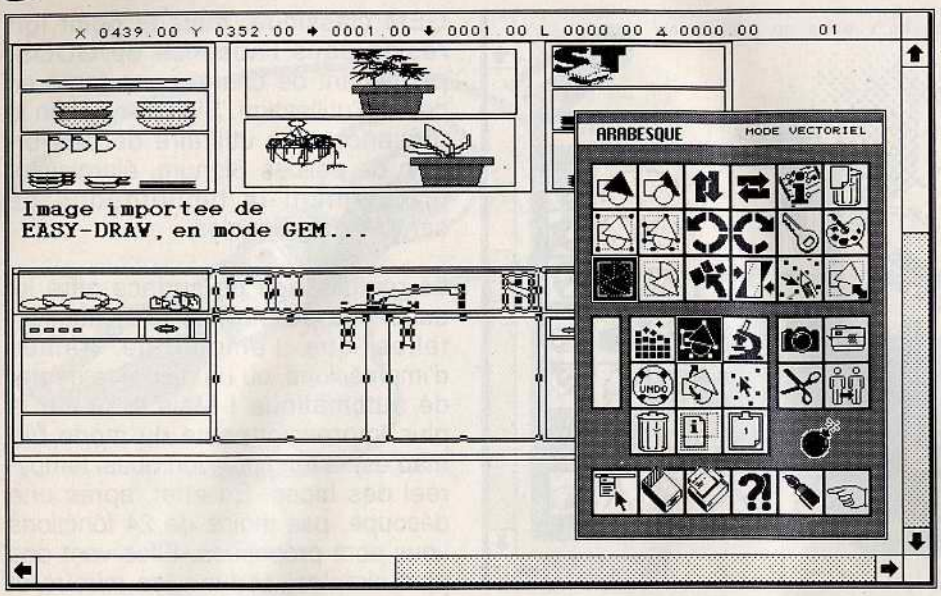
Pour conclure cette partie, apprenez qu'une loupe variable est présente, et qu'elle permet d'utiliser toutes les fonctions de la boîte à outils. Sont aussi implémentés une prévisualisation de la page avant impression, et un "clipping" protégant de toutes manipulations certaines de ses parties.

C'est une fois le dessin effectué que le mode vectoriel prend toute sa puissance. En effet, le logiciel est maintenant capable de dissocier un des éléments de l'image, et de lui faire subir les pires traitements. Couper, copier, coller, déplacer, retravailler, pivoter, incliner, épaissir, modifier la taille, envoyer dessus ou dessous, et bien d'autres fonctions sont au menu des réjouissances. Vous pouvez même grouper des objets de manière à les éditer simultanément.

L'image achevée, vous pouvez la transférer en bit-map ou l'imprimer

MODE VECTORIEL

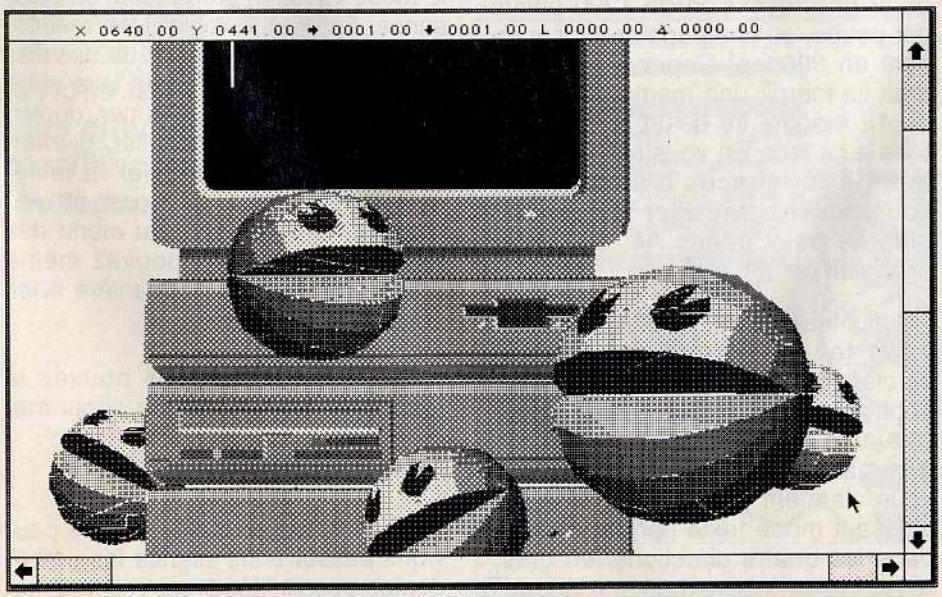
Enfin direz-vous, mais nous ne pouvions passer sous silence les précédentes possibilités !



Arabesque permet donc le dessin vectoriel, autrement dit, par objets. Nous pouvons résumer cette méthode en spécifiant qu'une image n'est plus ici mémorisée comme un ensemble de points (bit-map), mais comme une série d'objets (un objet étant lui-même composé de vecteurs mathématiques). Par exemple, un trait n'est plus une suite de points allumés, mais une droite avec un point de départ, un angle et une longueur.

Le principal avantage d'un tel système est qu'en utilisation normale, (soit en dessin), le logiciel affiche en 90 ppp (la résolution de l'écran), mais que lors de l'impression, il recalcule entièrement l'image de manière à produire une sortie parfaite (jusqu'à 1440 points par pouce de définition).

Les outils sont à peu près les mêmes qu'en bit-map, hormis les triangles et parallélogrammes 3D qui sont absents. Par contre, le mode texte est encore plus puissant puisqu'il permet l'alignement à gauche, à droite, le centrage ou la justification. Il peut même utiliser une police différente de celle affichée pour l'impression (cas des fontes Signum, avec plusieurs résolutions). Seuls regrets, l'absence de fontes vectorielles, et surtout des courbes de Bézier qui constituent tout de même aujourd'hui un "passage obligé" lorsqu'on parle de "vectoriel"...



Une image vectorielle

directement à partir de ce mode. Malheureusement, dans l'autre sens, la vectorisation de dessins n'est pas offerte (elle demande d'énormes calculs, mais peut-être dans une future version...), seul le transfert de bloc est possible, produisant un simple objet.

IMPRESSION

A ce jour, Arabesque peut gérer les 9 et 24 aiguilles (EPSON FX80, STAR NL10, NEC P6/P7...), les compatibles HP LaserJet (y compris la DeskJet) et la laser SLM 804. Dans le cas d'une imprimante à aiguilles, vous pouvez vous-mêmes la paramétrer grâce à une partie du mode d'emploi.

Avant le lancement de l'impression le positionnement de l'image sur la page est possible de manière à regrouper plusieurs écrans. Et enfin une rotation de 90 degrés permet de choisir son orientation.

COMPATIBILITÉ DES FICHIERS

Hormis les deux nouveaux formats qu'introduit ce logiciel, sont implémentés Degas (y compris en compressé), Doodle, Draw 3, Neochrome, Stad, Star-Painter, ainsi que les IMG, GEM et IFF (Amiga). Bonne nouvelle pour les utilisateurs de Publishing Partner Master, puisqu'ils trouveront sur la disquette un pilote d'importation Arabesque.

Nous avons tenté, en ces quelques lignes, de faire un tour complet des énormes possibilités d'Arabesque, mais ce dernier vous réserve de nombreuses découvertes et de longues heures d'utilisation. Un grand applaudissement aux développeurs de ce produit, qui ont réalisé cette fois un regroupement inédit et réussi de deux catégories de logiciels de dessin.

Sébastien Mougey

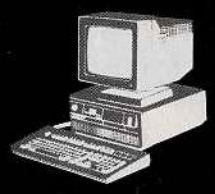
LIBRE OU OCCUPÉ PASSEZ AU SALON

2ème SALON DE LA **MICRO**

26-29 OCTOBRE 1990

ESPACE NORD - LA VILLETTE

la Villette — la grande halle

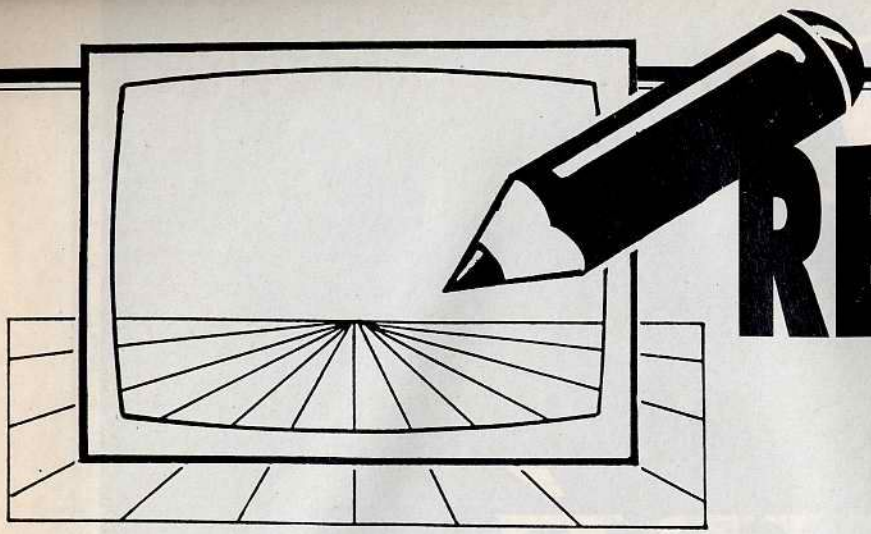


LA MICRO VRAIMENT INDIVIDUELLE

RFM LA RADIO FM 103.9

Pour tout savoir sur le salon.
Pour gagner de superbes cadeaux.
Pour se procurer une entrée privilégiée.

RFM LA RADIO FM 103.9



RETOUCHE

Les logiciels de traitement de l'image ne sont pas légion sur le ST. Pourtant, cette année, au Cebit, quantité de tels produits ont été présentés, tant pour l'image monochrome que couleur. La faiblesse graphique du ST étant ce qu'elle est, la plupart de ces nouveaux programmes nécessitent une carte graphique pour donner le meilleur d'eux-mêmes. Retouche, distribué par ALM, permet le travail sur l'écran monochrome sur ST, à l'aide du tramage, un artifice qui permet de simuler les niveaux de gris. Voyons donc ce qu'il en est.

La chaîne de microédition éditoriale ne saurait être complète sans un tel logiciel. En effet, il est rare que toutes les photos soit homogènes au niveau contraste et luminosité, ou que certaines ne présentent pas de défaut, tels des points blancs dans les zones noires, ou vice versa. C'est donc l'objet d'un programme comme Retouche que d'assurer toutes les corrections pour que l'image imprimée soit parfaite. Ce qui n'empêchera personne de l'utiliser dans une optique plus créative, grâce aux nombreux effets spéciaux proposés.

LE PACKAGE

En cette période estivale, les valisettes sont de saison, et c'est dans un tel emballage que vous découvrirez Retouche, après un allègement de près de 1.500 F. Les disquettes sont dans une enveloppe marron, information qui journalistiquement ne présente aucun intérêt. En revanche, vous trouverez sur ladite enveloppe un numéro de série que vous communiquerez à ALM, le distributeur, pour recevoir les indispensables accessoires du programme. En effet, ces derniers sont sérialisés, et ce n'est qu'une semaine plus tard que vous les recevrez, directement issus d'Allemagne. La solution est un peu lourde, et ALM est sur le point d'y trouver une alternative moins pénalisante.

Outre le programme, les quatre disquettes comportent plusieurs millions d'octets d'images compactées, qui servent de base de travail au tutoriel, un manuel de près de cent pages extrêmement bien fait. Il présente exhaustivement toutes les possibilités du logiciel au travers de nombreux exemples, et complète, par ses explications le manuel de

L'interface de travail

référence. Hélas ! la qualité de la traduction rend parfois les explications un peu confuses. Mais celui qui connaît le traitement de l'image naviguera sans problèmes au travers des quelques dérives syntaxiques.

Avant de quitter le magasin, la valise à la main, demandez quelques explications sur le décompactage des fichiers, car il n'en est dit mot dans la documentation, l'éditeur supposant que la manipulation du décompacteur ARCHIVE fait partie de la culture générale de chacun.

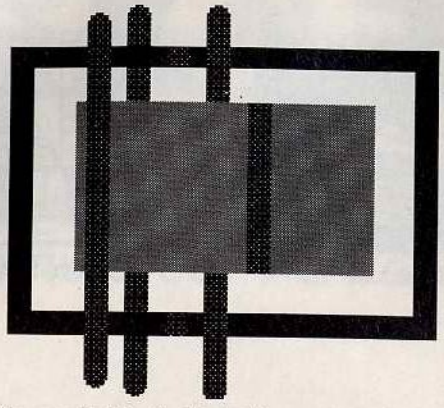
L'INTERFACE

Si les interfaces d'Outre-Rhin sont souvent en dehors des normes de convivialité auxquelles nous tenons, force est de reconnaître qu'hormis quelques maladresses de peu d'importance, la prise en main est plutôt immédiate, mais ne dispense nullement d'une lecture attentive des manuels. La finesse des corrections permises, les règles du jeu qui s'appliquent aux boutons gauche et droit de la souris, l'ensemble des raccourcis-clavier et l'influence des touches Shift, Control et Alternate sur les commandes ne se devinent pas.

L'écran de travail présente dans sa partie inférieure une longue barre qui contient la palette des gris, ou plutôt la palette des trames de gris. Le logiciel ne fonctionne qu'en monochrome, et c'est à la rentrée qu'une version Pro offrira les couleurs (256) et donc les niveaux de gris. Pour l'instant, seul un accessoire "hard" permet une représentation à la fois tramée et colorée sur un moniteur couleur. Cette palette comporte deux curseurs, l'un associé aux outils de dessin, l'autre à la gomme (le niveau de gris qui apparaît lorsque l'on efface une partie de l'image). On les positionne à la souris, ou plus pertinemment en "capturant" une teinte dans l'image à l'aide d'un Shift-Clic.

LES FICHIERS

Retouche ajoute encore un format de fichiers à une collection déjà bien étendue. Le format RTC sauvegarde l'image à la résolution de 640x400, après l'avoir compressée. En importation, c'est la totalité des formats Degas, plus ceux de Néochrome, Art Director, Stad, Doodle et Print Technic qui sont reconnus. Les couleurs sont transformées en niveaux de gris selon trois méthodes. Les deux premières donnent des résultats réalistes, tandis que la dernière est plus fantaisiste, et sera donc réservée à la création artistique. Curieusement, le TIF, format quasi universel des images à niveaux de gris (et couleur dans sa version 5) ne fait pas partie de cette liste. Rassurez-vous (mais étiez-vous réellement inquiets ?), il est cependant reconnu en tant que bloc (sa taille ne doit cependant pas dépasser les 640x400).



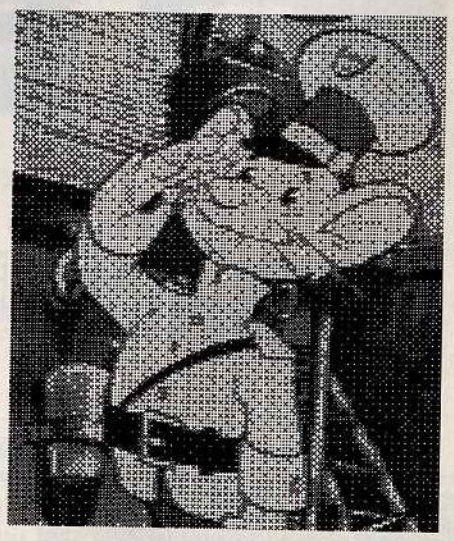
En mode Dessin, le trait recouvre toute l'image, les teintes plus claires ou plus sombres que le fond, toutes les teintes sauf le fond, et enfin, le fond seul.

L'exportation nous propose les mêmes formats (avec la même curiosité pour le TIF), mais aussi l'IMG. Nous avons poussé la curiosité jusqu'à placer un fichier TIFF dans PageMaker 3, sur un PC, et cela marche d'ailleurs très bien. On pourrait même, si ce n'était la limitation de taille, envisager d'employer le couple ST/Retouche pour traiter l'image, de manière très économique, dans un contexte éditorial sur PC ou Macintosh.

Les accessoires d'impression n'étant pas disponibles, et non francisés de toute manière, nous ne pouvons porter de jugement à leur égard. Attendons la rentrée de septembre pour les tester, avec, pourquoi pas, la version professionnelle. Le seul qui est fourni d'office concerne les imprimantes 9 aiguilles. Les autres sont dédiés aux 24 aiguilles, aux lasers PostScript (avec sortie de fichiers EPS), Hewlett-Packard (et compatibles) et Atari (SLM804). Le report, au niveau des accessoires, des fonctions périphériques - impression et numérisation - est tout à fait judicieux. Il permet, non seulement, de garder au programme une taille raisonnable, mais aussi d'éviter d'incessants changements de versions au fur et à mesure de l'apparition de nouveaux matériels.

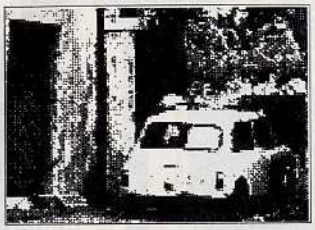
LA CORRECTION DE L'IMAGE

Il existe un accessoire, que nous n'avons pas eu, qui permet de numériser directement depuis le logiciel avec le scanner couleur d'Epson, le GT4000 (d'autres sont

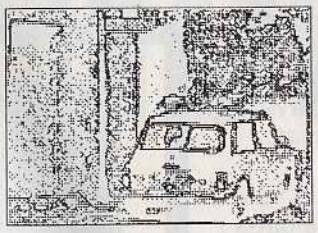


Travail sur une image peu contrastée pour faire ressortir le personnage, sans procéder à un détourage, qui élimine le fond.

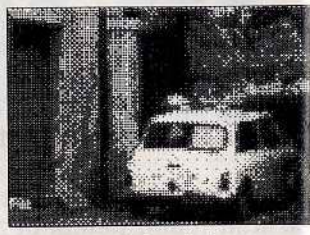
Contraster



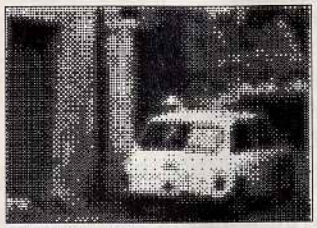
Contraste



Structure

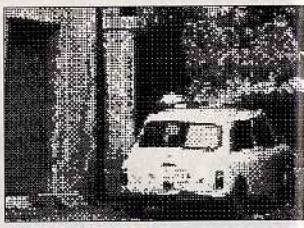


Adoucir 1

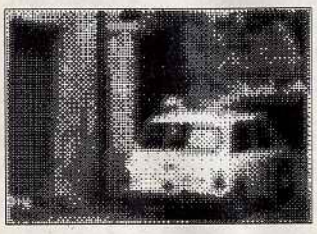


Adoucir 2

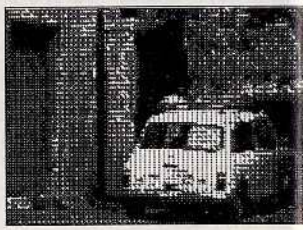
Image d'origine



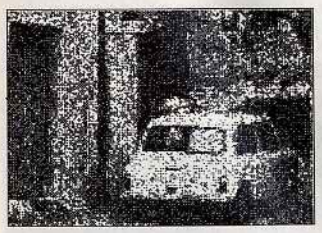
Eclaircir



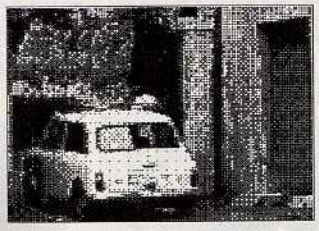
Délaver



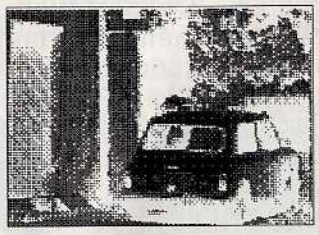
Structure



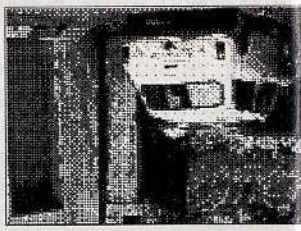
Grossier



Miroir vertical



Inversion



Miroir horizontal

en préparation). Sinon, on peut employer un digitaliseur vidéo ou charger une image issue d'un logiciel de dessin.

On distingue trois niveaux de correction. Ils s'appliquent à l'intégralité de l'image, à un bloc ou à une zone non masquée (le masque, est-il nécessaire de le préciser ?... est un masque qui protège certaines parties des modifications !).

Le premier niveau s'applique à l'image en général, et vise à ajuster la luminosité et le contraste. Si la fenêtre de dialogue appropriée propose deux curseurs pour cela, elle offre aussi un réglage via l'ajustement de la courbe Gamma, qui définit comment doit être transformé chaque niveau de gris. En l'absence de toute correction, c'est une droite ascendante, inclinée à 45• vers la droite. Un basculement vers l'horizontalité ou la verticalité correspond à une diminution ou un accroissement du contraste, tandis qu'un déplacement de la droite vers le coin supérieur gauche ou le coin inférieur

Comparer C'est Choisir

SUPERCHARGER®
L'émulateur PC pour Atari ST

Livré en standard avec 1Mo de RAM sans augmentation de prix

Supercharger est la référence en matière d'émulation PC. Il se présente sous la forme d'un boîtier externe de très belle qualité ne nécessitant **aucune intervention** à l'intérieur de votre ordinateur, vous ne perdez donc pas votre **garantie**. Celui-ci s'installe en **quelques instants** et vous permet d'utiliser la plus grande partie des logiciels pour compatibles, et ce à une vitesse très largement supérieure à celle d'un PC XT. De plus, du fait de sa conception, il devient très vite un PC transportable d'un ordinateur Atari à l'autre®.

DESCRIPTION :

- Processeur Nec V-30 cadencé à 8Mhz
- 512 Ko de mémoire vive (extensible à 1Mo)
- Support de co-processeur arithmétique 8087
- Emulation CGA, Hercules
- Connection sur le port DMA (disque dur) sans monopoliser l'accès
- Compatibilité Hypercache
- Emulation à 100% des ports parallèle et série (→19200 bauds)
- Installation sur tous les disques durs, gérant simultanément sous MS-DOS 18 partitions
- Reconnaissance d'un lecteur externe 5"1/4 ou 3"1/2
- Indice Norton 4.2
- Test de performances Norton :
  -Pc-xt(8088)4.77Mhz: 100%
  -SUPERCHARGER: 297%
- Reconnaissance automatique des formatages disquette et disque dur Atari® (facilite les transferts)
- Impression sur toutes les imprimantes parallèles ainsi que sur Laser Atari
- Livré avec un **driver pilotant la souris Atari**

LE SEUL À VOUS PROPOSER

►Un **MS-DOS® 4.01 en français**

►Une **fonction HOTKEY** vous permettant de basculer entre le monde PC et Atari par une simple combinaison de touches) et cela sans perdre l'application en cours sous MS-DOS

►Un **RAM disque** qui s'installe sous **TOS**, qui résiste au Reset, qui s'utilise aussi bien sous Atari que sous MS-DOS et conserve les données lors du Switch entre PC et ST

►Un **RAM disque** qui s'installe sur **Supercharger**, utilise la mémoire de l'émulateur et résiste au Reset

►Un **fonctionnement parfait** sur tous les Atari ST®, STE®, Mega ST®, **STACY® et TT®** (testé!)

►Une **TOOLBOX**

En fait, des outils de développement qui vous permettent sous Atari de programmer Supercharger et de faire cohabiter en calcul le 68000 du ST et le NEC V30 de l'émulateur: une excellente approche de la programmation parallèle

En vente chez tous les bons distributeurs. Supercharger est un produit BETA System

A.L.M. 1, Rue Pierre Dupont — 93200 Saint-Denis — France
Tél / Fax : 16 (1) 42.43.36.95

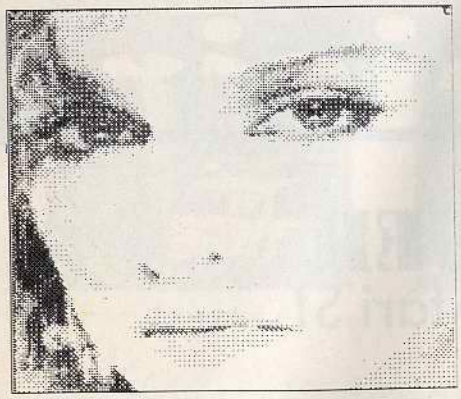
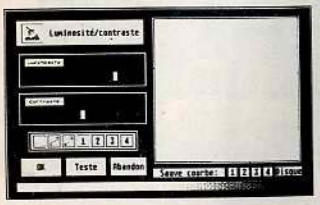


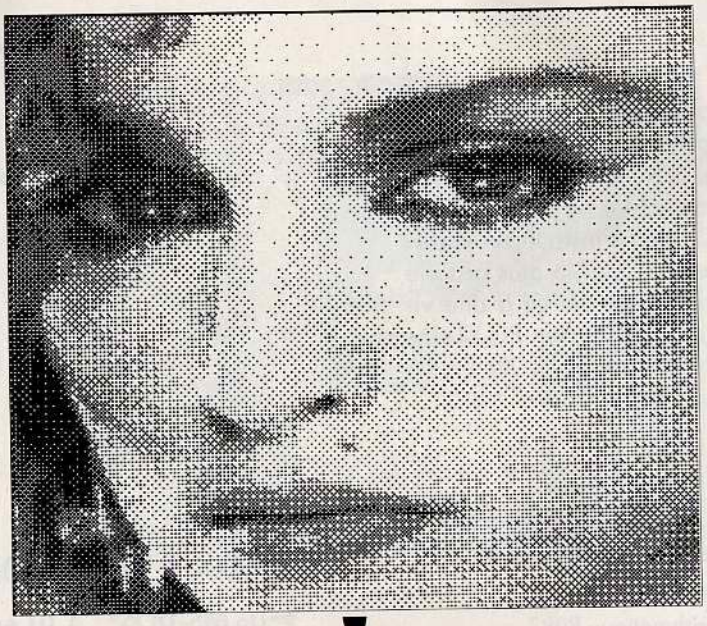
Image plus claire

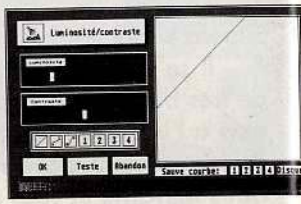
Courbe normale

Image plus sombre

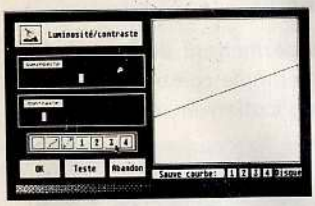


Courbe plus claire

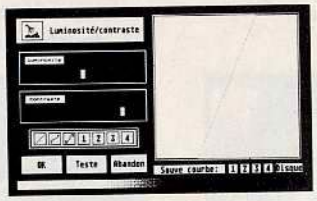




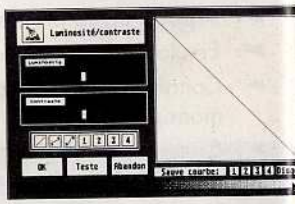
Courbe plus sombre



Courbe moins contrastée



Courbe plus contrastée



Courbe négative

Image moins contrastée

Image plus contrastée

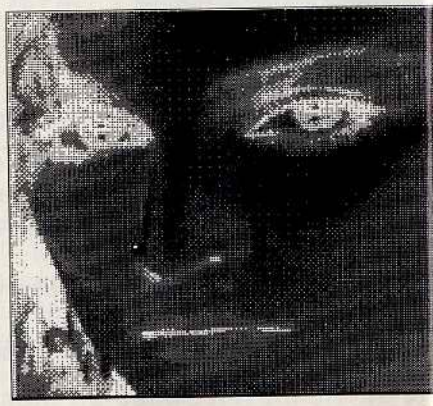


Image négative

droit l'assombrit ou l'éclaircit. Cela paraît compliqué, mais c'est logique puisque les abscisses représentent les gris de départ, et les ordonnées les gris d'arrivée. Par exemple, si la droite occupe l'autre diagonale, cela signifie que le blanc (0) donne du noir (255), et inversement. Ce qui correspond à une inversion de l'image qui passe en négatif.

La correction ne s'effectue pas en direct, aussi un bouton Test permet de juger du résultat avant la validation par OK. On peut sauvegarder quatre courbes dans les Préférences, ou indépendamment sur disque.

Le deuxième niveau vise à corriger l'image dans son apparence, pour la rendre plus ou moins subjective. Les effets vont de la simple augmentation du contraste à l'attribution d'une texture toilée. Nous avons, chanceux que vous êtes, représenté tous les effets du logiciels pour vous épargner une fastidieuse énumération descriptive. En revanche, vous n'échapperez pas au buffer Undo. Vous vous doutez bien que corriger une image est souvent une succession d'essais, et qu'il est bien agréable de pouvoir revenir à l'image de départ si une tentative s'avère infructueuse. À cet effet, le logiciel, si la mémoire le permet, installe un tampon de 256 K, dans lequel vous pouvez sauvegarder l'image avant toute modification (dans certains cas, cette sauvegarde est lancée automatiquement) par appui sur la barre d'espacement. La touche Undo intervertit les deux images, ce qui permet de comparer rapidement la version initiale et celle corrigée.

Les corrections des deux premiers niveaux s'appliquent aussi à la partie de l'image non masquée, ou à un bloc (délimité à la souris).

Le troisième niveau correspond à des modifications locales, qui permettent de mettre en valeur une partie de l'image ou d'apporter des changements créatifs. Pour ce faire, on dispose d'outils de dessin, comme le Crayon, la Craie, le Pinceau ou bien le tampon, qui permet de simuler les trois outils précités. Le menu Paramétrage en assure le réglage.

D'autres outils, comme la Goutte d'eau ou le Doigt, permettent respectivement d'estomper ou d'étaler des teintes. Là aussi, le paramétrage assure une action plus ou moins efficace, pour un niveau de correction plus ou moins subtil. Le Rectangle assure, en outre, la production de dégradés. On passera très vite sur la fonction de remplissage pour s'attacher aux blocs. Leur définition et redimensionnement sont on ne peut plus traditionnels. En revanche, la copie et le déplacement donnent des effets différents selon la combinaison de touches employée. Ainsi, avec Control, les valeurs de gris du blocs sont-elles soustraites de celles de l'image, tandis qu'avec Alternate, c'est la valeur moyenne des pixels correspondant du bloc et de l'image qui est retenue. Il existe d'autres combinaisons qui permettent de moduler les effets. On le voit, tout semble possible et il est vivement conseillé d'expérimenter à fond les effets pour savoir, en fonction de chaque cas de figure, quel outil ou quelle combinaison d'outils est le plus approprié.

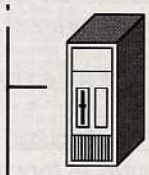
Réseau Biodata

OU COMMENT RELIER VOS ORDINATEURS SOUS ETHERNET

Le réseau Biodata, c'est avant tout un partage de ressources, une mise en commun de fichiers et une seule mémoire de masse pour un ensemble d'utilisateurs.

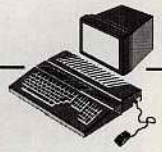
Biodata, c'est ensuite une convivialité, une souplesse d'utilisation et une rapidité de transfert (10MBit/s)

Biodata, c'est enfin le moyen d'intégrer un ordinateur Atari à tout réseau hétérogène fonctionnant sous Ethernet (PC, Unix...)



Serveur MS-DOS
- Jusqu'à 254 utilisateurs
- 2048 MB max. sur disque dur
- Poste dédié ou non
- Possibilité de multiplier les serveurs en parallèle pour un gain en fiabilité

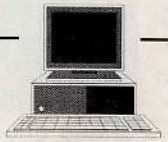
Serveur ATARI
- Jusqu'à 4 utilisateurs
- 16 MB max. sur disque dur
- Poste non dédié
- Un seul serveur possible



Utilisateur ATARI

Sous mode Atari
- 8 Ko de buffer
- NSS, Boîte aux lettres, spooler

Sous mode Unix
- 32 ko de buffer
- TCP/IP, Autoboot possible
- NSS, Boîte aux lettres, Spooler

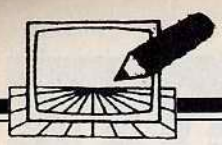


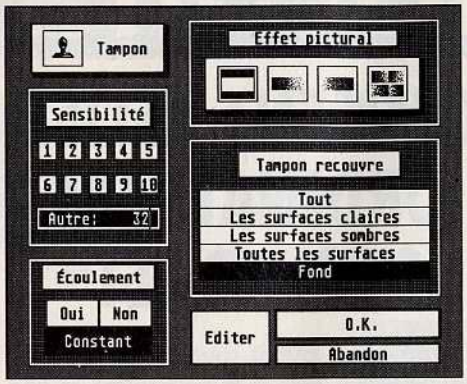
Utilisateur MS-DOS
- 8 ko de buffer
- NSS

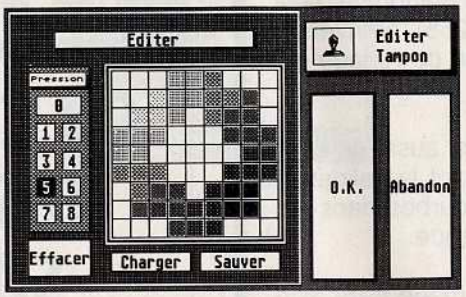
Passerelle — **DEC Net – NFS/SUN – Novell Unix – MacIntosh – etc...**

Biodata et Bionet sont des marques déposées de Biodata GmbH.
Toutes les autres marques citées sont déposées par leurs propriétaires respectifs

A.L.M. 1, Rue Pierre Dupont
93200 Saint-Denis
Tél: 16 (1) 42.43.36.95







Le réglage du tampon et la boîte d'édition de sa forme. L'option Recouvre définit à quels niveaux de gris s'applique l'outil. Les surfaces claires ou sombres sont définies par rapport à la couleur du fond (celle qui apparaît lorsque l'on utilise la gomme).

Si l'emploi de la gomme laisse apparaître la couleur du fond, son usage avec l'option Restaurer est très spectaculaire, puisque ce qui remplace les parties effacées est pris dans le buffer Undo. Cela ouvre la porte à des effets des plus réussis.

LE MASQUAGE

La pose d'un masque est manuelle ou automatique. Dans le premier cas, on passe en mode Masqueur. Le bouton gauche trace le masque, et le droit l'efface. Clair et net. Si on encadre une partie de l'image, un Shift droit-Clic permet de remplir la surface ainsi délimitée pour en faire un masque plein. Le mode automatique est plus subtil, puisqu'il consiste à masquer les parties de l'images selon des critères particuliers : plus clair ou sombre que la couleur du fond, par exemple. Comme il est parfois plus facile de masquer la zone à traiter, que celle à protéger, une fonction "Inverser masque" facilite le travail.

On peut ainsi placer des masques discontinus et faire de la microchirurgie correctrice, voire pallier aux fonctions absentes. Par exemple, sur Macintosh, on trouve une fonction Déparasiter, qui consiste à nettoyer les surfaces blanches de "pétouilles" noires, correspondant à des minis-taches. En combinant l'option Masque avec d'autres fonctions, on pourra débarrasser les aplats de blanc des taches noires (ou l'inverse).

LA CRÉATIVITÉ

La créativité, c'est employer tous les outils et fonctions, non pas pour restituer l'image dans toute son objectivité, mais, au contraire, pour en créer une nouvelle en dehors des contingences du réel, selon les impératifs flous du rêve et de l'imaginaire.

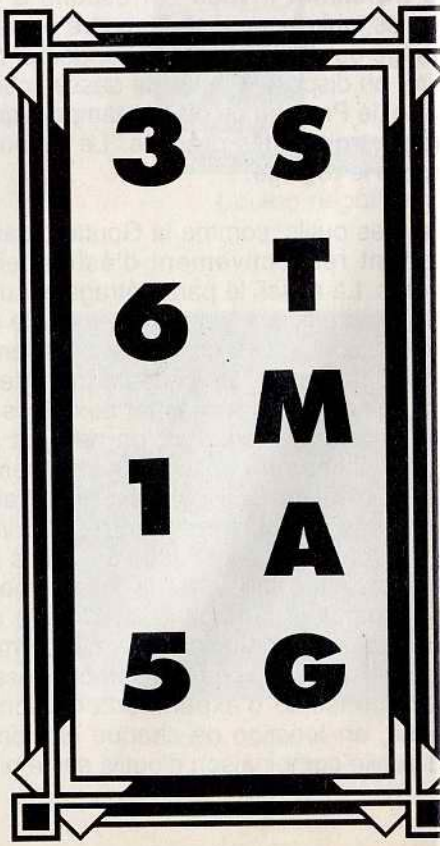
Outre la débauche fonctionnelle déjà examinée, Retouche, via le buffer Undo, permet de mélanger deux images différentes, avec un réglage, en pourcentage, de cette fusion. Une fonction similaire, Évaluer, permet de réaliser des estompes progressives de l'image (lorsque le buffer Undo contient un Dégradé). Le manuel d'apprentissage est d'ailleurs une mine d'or qui explique les mille et uns usages du logiciel, comme par exemple l'incrustation d'une image dans le contour d'une lettre (si le logiciel n'offre pas de fonction Texte, on pourra néanmoins employer des fichiers TIF, fournis sur les disquettes, qui correspondent à plusieurs polices).

CONCLUSION PROVISOIRE

Malgré la profusion, le testeur jamais satisfait trouvera toujours à redire. Il existe deux ou trois fonctions que l'on aimerait voir. Tout d'abord, la définition d'un masque par courbes de Bézier serait plus précis et efficace qu'un simple tracé à main levée. Enfin, si la copie d'un bloc s'accompagnait d'une sorte d'anti-aliasing, afin que le bloc se fonde dans son nouveau décor sans attirer l'attention, notre bonheur serait complet (voyez la couverture du dernier ST Mag et vous comprendrez l'utilité de la chose).

En fait, les principales limitations résident dans le mode d'affichage standard du ST, qui est loin de satisfaire tous le monde. Un écran à véritables niveaux de gris est un confort quasi indispensable pour employer au mieux Retouche, dont cette première version nous a surpris par l'étendue des possibilités. Aussi attendrons-nous la version Professionnelle, prévue pour la rentrée, pour porter un jugement définitif. Nous en profiterons aussi pour évaluer les fonctions d'impressions. Mais d'ores et déjà, l'examen est réussi... et avec mention.

Laurent "L.K." KATZ



3615 JESSICO
CPC-ST-PC-AMIGA

JESSICO

Quand les prix sont si bas, les souris dansent !

LOGICIELS JEUX

	ST	PC	AMIGA
COLLECT. PLATINIUM 2	239		
COMPILATION 1	239		249
PREMIER COLLECT. 3	289		
TRIAD 3	289		289
ACTION D'ENFER	239	239	239
LES GEN D'OR	239		239
LES JUSTICIERS	239		239
MAGNUM 4	289		389
LES VAINQUEURS	289		289
OCEAN 5 STARS	239		239
PRECIOUS METAL	239		239
SIMULATION HITS	239	239	
STAR WARS TRILOGY	189		
SUPER QUINTET	249		
ALPHA WAVES	249	249	249
ANTAGO	192		192
BANGKOK KNIGHT	192		
BAT + CARTE	379		
BATTLE OF BRITAIN	239	289	239
BATTLEHAWKS	239	239	239
BATTLE MASTER	289		289
BLACK TIGER	192		239
BLADE WARRIOR	239	239	239
BLOCK OUT	239		
BOMBER	289	339	289
BLOODWYCH	239		239
BLOODWYCH DATA DISK	165		
BLUE ANGEL	239		239
BUBBLE +	219	219	219
BUDOKAN			239
CABAL	192		239
CARMEN SAN DIEGO		309	
CASTLE MASTER	239	289	239
CHAOS STRIKES BACK	219		
CHUCK YEAGER 2 0	242		242
COLORADO	239	259	239
COMBO RACER	192		239
COSMOS	239		239
CRACK DOWNS	192		239
CYBERBALL	192	239	192
DAN DARE 3	192	229	192
DARK SAT	239		
DOUBLE DRAGON 2	192	239	192
DRAGON BREATH	289		
DRAGON FLIGHT	239		
DRAGON SPIRIT	239	239	192
DRAKKHEN	289	289	289
DRIVIN FORCE	259		
DYNASTY WARS	192		242
DYTER 07	239		
EAGLE S RIDER	239	239	239
E-MOTION	192	239	239
ESCAPE FROM PLANET	192	229	192
F 29	229		239
F15 STRIKE EAGLE 2		339	
FALCON	232		289
FALCON MISSION DISK	192		192
FATAL GRAVITY	239		
FIRE AND FORGET 2	259	259	259
FLOOD	242		242
GHOSBUSTERS 2	239	329	239
GRAVITY	239		
GREAT COURTS	239		239
HAMMERFIST	239		239
HARD DRIVIN	192	239	192
HARRICANA	239		239
HEAVY METAL	192		239
HEROES QUEST	329		
HIGWAY PATROL 2	239	239	239
HILLSFAR	239	239	
HOT ROD	239		239
ICE YACHTS	239	239	239
IMPOSAMOLE	192		192
INFESTATION	239		239
INTER 3D TENNIS	242		242
ITALY 1990	239		239
IVANHOE	229		239
KICK OFF	179		259
KICK OFF . PROLONGATIO	149		
KID GLOVES	239		239
KLAX	192	239	192
KNITGHT FORCE	179	259	259
LE MANOIR DE MORTEVIL	192		
LIVERPOOL	192		192
LODE RUNNER	192	192	
LOST PATROL	192		239
MANCHESTER UNITED	192		239
MATRIX MARAUDERS	192	239	
MAUPITI ISLAND	279		279
MINDWINTER	289	329	
NEVERMIND	192	239	192
NINJA SPIRIT	239		239
NINJA WARRIORS	192		199
OCEAN BEACH VOLLEY	192		239
OPERATION STEALTH	289		
PINBALL MAGIC	192	192	192
PIPEMANIA	259	259	259
PIRATES	259		239
PLAYER MANAGER	259		259
POPULOUS	239	239	239
POWER BOAT	239	239	239
PROJECTYLE	242		
RAINBOW ISLAND	192		239
ROCK STAR	239	239	239
SHADOW WARRIORS	192		239
SHERMAN M4	239	239	
SIM CITY	289	289	289
SKIDOO	239		
SKIDZ	192		192
SONIC BOOM	239		
SPACE HARRIER 2	239		239
SPIDERMAN	259		259
STAR BLADE	239	259	239
STARLIGHT	239		
STRYX	192		192
SUPER CARS	192		192
SUPER WONDERBOY	192		239
TENNIS CUP	239		239
THE STRIDER	192		239
THE THIRD COURIER	239		239
TURRICAN	242		242
ULTIMA V	279		
ULTIMATE DARTS	192		
ULTIMATE GOLF	239	239	239
VAUX	239		239
VENUS	192		192
VROOM	192		192
WARHEAD	192		239
WATERLOO	239	239	239
WILD STREETS	259	259	259
WORLD CUP SOCCER	192		192
X OUT	192		192
XENON2 MEGABLAST	239	239	239
XIPLOS	239	239	239
ZAC MAC CRACKEN	239		

EDUCATIFS

ALLEMAND 4/3e	220	240	
ALLEMAND PRIMAIRE	220	220	
ANGLAIS PRIMAIRE	220	220	
ESPAGNOL PRIMAIRE	220	220	
ITALIEN PRIMAIRE	220	220	
HISTOIRE PRIMAIRE	199		
SCIENCE PRIMAIRE	220	220	
PLANETE CONNAISSANCE	199		
ANGLAIS 4/3e	240	240	
ANGLAIS CONFIRME	245	245	
ANGLAIS DEBUTANT	245	245	
ANGLAIS PERFECTION. 1er	245	245	
ANGLAIS PERFECTION. 2e	245	245	
ANGLAIS POUR LE BAC	460	460	
ANGLAIS TOP NIVEAU 2/1er	245	245	
APPRENDS MOI A LIRE 1	245	245	
APPRENDS MOI A LIRE 2	245	245	
BALADE A COLOGNE	250	250	
BALADE A SEVILLE	250	250	
BALADE A BIG BEN V.3	250	245	250
BALADE PAYS DE BIG BEN 6/5e	250		
BOSSE DES MATHS 1er	245	245	245
BOSSE DES MATHS 2e	245	245	245
BOSSE DES MATHS 3e	220	220	220
BOSSE DES MATHS 4e	220	220	220
BOSSE DES MATHS 5e	220	220	220
BOSSE DES MATHS 6e	220	220	
CALCUL PRIMAIRE	215		
CODE FACILE	230	230	
ECRIRE SANS FAUTE VOL 1	245	245	245
ECRIRE SANS FAUTE VOL 2	245	245	
EDUC-MATERNELLE 1	220	240	
EDUC-MATERNELLE 2	220	240	
EDUC.PRIM. CALCUL CE1-CM2	220		
ENGLISH FOR BUSINESS	490	490	
ENIGME A MADRID 4e 3e	250	250	
ENIGME A MUNICH 4e 3e	250	250	
ENIGME A OXFORD 4e 3e	250	250	
FRANÇAIS CM	220		
FRANÇAIS SONS CP/CE	220	240	
GEOMETRIE PLANE	220	240	
GRAMMAIRE 6/5e	220	240	
LECTURE CP	220	240	
MATHS 1er	240	240	
MATHS 2e	240	240	
MATHS 3e	220	240	
MATHS 5e	220	240	
MATHS 6e	220	240	
MATHS CE	220	240	
MATHS CM	240	240	
MICRO BAC ALLEMAND 1/TER	220	220	
MICRO BAC ANGLAIS 1/TER	225	220	
MICRO BAC ESPAGNOL 1/TER	225	220	
MICRO BAC FRANCAIS 1/TER	225	220	
MICRO BAC GEO 1/TER	225	220	
MICRO BAC HISTOIRE 1/TER	225	220	
MICRO BAC MATHS B 1/TER	225	220	
MICRO BAC MATHS C+E 1/TER	225	220	
MICRO BAC MATHS D 1/TER	225	225	
MICRO BAC PHYSIQUE/CHIMIE	225	220	
MICRO BREVET ALGEBRE	220	220	
MICRO BREVET FRANCAIS	220	220	
MICRO BREVET GEOGRAPHIE	220	220	
MICRO BREVET GEOMETRIE	220	220	
MICRO BREVET HISTOIRE	220	220	
MULTICOURS 3e	245	245	
MULTICOURS 4e	245	245	
MULTICOURS 5e	245	245	
MULTICOURS 6e	245	245	
ORTHOGRAPHE CM	220	240	
TOP NIVEAU ANGLAIS 2/1er	250		
VISA POUR HYDE PARK 6e	250	250	

PRIX BUDGET

ACTION SERVICE	95		95
ARCHE CAPTAIN BLOOD	95		
ARTIFOX	95	95	95
BARBARIAN	95		95
BARD'S TALE 1	95	95	95
BIONIC COMMANDO	95	95	95
BOBO	95		
DEEP SPACE	95		
EMPIRE	95	95	95
FAST FREAK		95	
FUSION	95		95
HOSTAGES	95		95
LEADERBOARD	95	95	95
LEGACY OF THE ANCIENTS		95	
MARBLE MADNESS	95	95	95
MUSIC CONSTRUCTION SET	95		
OUT RUN	95		95
PHM PEGASUS		95	
RACK EM		95	
SERVE AND VOLLEY		95	

UTILITAIRES ST

ATLAS 2	250
A. DEBOG	590
AUTOFORM. BASIC GFA	285
AUTOFORM. STOS BASIC	275
BECKER DAO	**975**
BECKER TEXT2	725
CALAMUS	**2350**
CALCOMAT II	575
CALLIGRAPHEUR PRO	1490
CLAVISSIMO	280
Copieur SYNCHRO Express	**495**
CRAFT 2	690
CYBER PAINT 2.0	690
DACTYL	250
DEGAS ELITE	**245**
DEVPAC V.2	730
DIGITALISEUR VIDI ST	**990**
DISCOCOPIE	**490**
EASY GEM	**265**
FACT.+ STOCK FACILES	**395**
FLEXIDISC	**250**
FLEXIDUMP	**390**
FM MELODIE MAKER	**795**
FUN FACE	**395**
GESTION BUDGET PERS.	300
GESTOCK 90	1775
GFA ASSEMBLEUR	575
GFA BASIC 3.0	690
GFA COMPILATEUR 3.0	325
GRAPHIC CITY	295
HARD DRIVE TOOL KIT	299
HORLOGE TIMEKEEPER	325
HOTWIRE	490
INTERPRETEUR C 2.0	599
K SPREAD 4	990
LE COMPTABLE 2	775
LE GESTIONNAIRE	585
L'ELECTRONICIEN	975
LE REDACTEUR 1.98	575
MASTERSOUND	**435**
MENTEL	**485**
MULTIDESK	**390**
PACK ANTIVIRUS ST	**199**
PACK GFA BASIC 3.0	**775**
PRINT MASTER+	350
PROFIMAT	470
PROSCORE	1675
QUARTET	**495**
PROTOS	**250**
REVOLVER	**425**
SPACK	**450**
SCRABBLEUR	250
ST COMPTES	**290**
ST REPLAY V.4.10 FR	**690**
ST REPLAY PRO	1375
STOS BASIC	**385**
STOS COMPILER	**249**
STOS MAESTRO	**269**
STOS MAESTRO+	**699**
STOS MUSICIEN	**215**
STOS SPRITES	**179**
SUPERBASE	**595**
SUPERBASE 2	**890**
SUPER TOOLKIT II	**199**
TEXTOMAT	345
TURBO ST	390
WERCS	**325**
WORD UP 2.1	**675**
+ FONTZ	**285**
Z TIME	580
ZZ COM	475
ZZ DRAFT	**395**
ZZ IDEE	**495**
ZZ LAZY PAINT	810
ZZ ROUGH	**395**

DISQUETTE NETTOYAGE 3.50" : 59 F

UTILITAIRES AMIGA

AEGIS ANIMATOR	825
AEGIS AUDIOMASTER II	625
AEGIS DRO 2000	2150
AEGIS IMPACT	725
AEGIS SONIX	575
AEGIS VIDEOSCAPE 3D	1150
AEGIS VIDEO TITLER	1025
AMAS	825
ANIMATE 3D	1250
COMIC SETTER	825
Copieur SYNCHRO Express	**495**
DATAMAT	495
DELUXE MUSIQUE	650
DELUXE PAINT 3	825
DELUXE VIDEO	650
DELUXE PRODUCTION	1750
DESIGN 3D	875
DIGIPAINT II	750
DIGITALISEUR VIDI	**990**
Dr T's	1975
DEVPACK 2.0 ASS	825
EXCELLENCE	2350
FANTAVISION	475
GFA BASIC	725
GRABBIT	275
HANDYSCAN. GENIUS	1850
INSTANT MUSIC	245
KING WORDS 2.0	525
K SEKAASS	725
MASTER SOUND	425
MAXIPLAN 500	1225
MAXIPLAN +	1750
MCC PASCAL 2	775
MODELEUR 3D	775
MUSIC STUDIO	310
PACK GFA BASIC	475
PAGE FLIPPER	875
PAGE FLIPPER + F/X	1325
PAGE SETTER	750
PERFECT SOUND 3.0	850
PHOTOLAB	1175
PHOTON PAINT 2.0	1175
PHOTOPAINT II	625
PRINT MASTER+	375
PRO WRITE	1025
PRO SAMPLER STUDIO	795
STUDIO 24	1175
SUPERBASE	595
SUPERBASE PRO	1475
SYNTHIA	825
TEXTCRAFT +	725
THE CALLIGRAPHER	790
TRACK 24	475
TV SHOW	875
ULTRA CARD PLUS	710
VIDEO GENERIC MASTER	775
VIDEO WIPE MASTER	875

JOYSTICKS

ADAPT. 4 JOYSTICK ST	75
ADAPT. 4 JOYSTICK AMIGA	75
PHASOR ONE (+ MONTRE)	109
COBRA	480
COMPETITION PRO	125
PRO 5000	125
QUICKJOY JUNIOR	85
QUICKJOY STICK	99
QUICKJOY 2	89
QUICKJOY 3 SUPERCHARGER	**105**
QUICKJOY 5 SUPERBOARD	**192**
QUICKJOY IR INFRAROUGE	**390**
QUICKJOY M5 IBM XT/AT	**195**
QUICKJOY VI JETFIGHTER	**145**
QUICKSHOT TURBO II	**135**
KONIX SPEED KING	109
KONIX SPEED KING AUTOFIRE	125
KONIX THE NAVIGATOR	145
JOYSTICK CHEETAH PC	149
CARTE PC	179
PISTOLET WEST PHASER	**339**
TURBO PEDALE	**269**

RUBANS POUR IMPRIMANTES

tous les rubans par 3 = 144 F

RUBAN CITIZEN 120D	59
RUBAN EPSON MX-80	59
RUBAN DMP 2000/2160	59
RUBAN DMP 3000/3160	59
RUBAN DMP 4000	59
RUBAN STAR NL10	59
RUBAN STAL LC10	59
RUBAN SM804	59

HOUSSES

Housse complète clavier+écran

HOUSSE PC1512 COUL./MONO	109
HOUSSE CITIZEN 120 D	79
HOUSSE STAR LN10	79
HOUSSE STAR LC10	79
HOUSSE ATARI ST	119
HOUSSE AMIGA	119
HOUSSE SMM 804	99

LIBRAIRIE

PC

102 PROGRAMMES PC	135
CLEFS POUR PC (DOS 3.3)	250
CLEFS POUR PC/AT	250
COMPAGNON DU DISQUE DUR	295
DOMPTEZ VOTRE SOURIS	149
GUIDE PETER NORTON	295
SUPER JEUX PC+COMPATIBLES	120
BIEN DEBUTER PC 1512	149
LE LIVRE DU BASIC 2	179
LE LIVRE DU BASIC 2 + D	229
TRUCS ET ASTUCES PC 1512	179
TRUCS ET ASTUCES PC 1512 + D.	229

ST

102 PROGRAMMES BASIC ST	145
102 PROGRAMMES GFA BASIC	135
ASSEMBLEUR 68000	145
ATARI ST EFFICACE	98
ATARI ST EN ACTION	135
ST EN FAMILLE - BASIC GFA	145
BASIC ST METHODE PRATIQUE	125
CLEFS POUR ATARI ST	195
CLEFS BASIC GFA 2 ET 3 SUR ST.	135
MUSIQUE SUR ATARI ST	135
PEINTRE ET MUSICIEN SUR ST	148
PROGRAMMER EN BASIC ST	125
SUPER JEUX ATARI ST BASIC GFA	140
BIEN DEBUTER AVEC ST/STE	129
BIEN DEBUTER GFA BASIC 2+3	129
BIEN DEBUTER LE REDACTEUR	129
PACK GRAND LIVRE ATARI ST	199
LE LIVRE DE CALAMUS	199
LIVRE GFA BASIC 3.0	199
LIVRE GFA BASIC 3.0+D	265
LIVRE DES IMPRIMANTES	149
LIVRE SUPERBASE (V.2+3 PRO	169
LE LIVRE DU LANGAGE MACHINE	149
MIDI MUSIQUE+SEQUENC.	99
TRUCS ET ASTUCES ATARI ST + D	299

AMIGA

102 PROGRAMMES AMIGA	135
AMIGA EN FAMILLE	145
BASIC GFA SUR AMIGA	98
CLEFS POUR AMIGA N.ED.	195
L'AMIGA EN ACTION	168
LIVRE DE L'AMIGA BASIC (LE)	145
SUPER JEUX AMIGA	140
BIEN DEBUTER AVEC L'AMIGA	149
BIEN DEBUTER EN C SUR AMIGA	149
SOS AMIGADOS/AMIGABASIC	149
LIVRE LECTEUR DISQUETTE+D	299
LA BIBLE DE L'AMIGA	299
LA BIBLE DE L'AMIGA + D	399
LE LIVRE DE LA MUSIQUE + D	199
LE LIVRE DU GRAPHISME	249
LE LIVRE DU GRAPHISME + D	349
GRAND LIVRE AMIGABASIC	249
GRAND LIVRE L'AMIGABASIC+D	349
LIVRE DE L'AMIGADOS	199
LIVRE SUPERBASE (V. PRO)	169
LIVRE DU GFA BASIC AMIGA	149
LIVRE DU GFA BASIC AMIGA + D	249
LIVRE DU LANGAGE MACHINE	199
TRUCS ASTUCES AMIGA - V.II	129
TRUCS ASTUCES AMIGA - V.II+D	229

ACCESSOIRES

COPY HOLDER	189
FILTRE ECRAN 14" Coul.	**169**
FILTRE ECRAN 14" Monoc.	159
SUPPORT IMPRIM. 80 Col.	149
SOURIS GENIUS AMIGA	279
SOURIS GENIUS ST	279
SCANNER GENISCAN GS4500 ST.	1790
GENISCAN GS4500 AMIGA	1790
GENISCAN GS4500 IBM	1690
INTERFACE MIDI AMIGA	249
LECTEUR EXT. 3.50" AMIGA	899
CABLE PERITEL AMIGA	129
LECTEUR INT.3.50" ST	790
LECTEUR EXT.3.50" NEC ST	990
CABLE PERITEL ST	139
EXTENSION 512K ST	799
EXT.AMIGA 512K	750

COPIEUR SYNCHRO EXPRESS

- Système de duplication de disquettes en moins de 30 secondes.
— SYNCHRO EXPRESS nécessite un deuxième lecteur.
— Sélection par menu des débuts et fins de pistes (jusqu'à 85 pistes), une ou deux faces, autosélect.
— Duplication d'autres formats tels IBM, MAC, etc.

PROBABLEMENT LE MEILLEUR COPIEUR !

SYNCHRO EXPRESS ST	369 F
SYNCHRO EXPRESS AMIGA	369 F

BOITIERS DE RANGEMENT

avec serrure + clé

BOITIER DD50L	50x5.25"	119
BOITIER DS100L	100x5.25"	149
BOITIER DS40LB	40x3.50"	99
BOITIER JSY80	80x3.50"	99

SUPER PROMO

SUPERBASE Pro+Dév. SUPERBASE	975
BECKERCALC 2+SUPERBASE 2	975
COPIEUR SYNCHRO EXPRESS	369
Pack GFA Basic+Compil+As.3.0	775
STOS BASIC	385
STOS COMPILER	249
STOS MAESTRO	269
STOS MAESTRO+	699
STOS MUSICIEN	215
STOS SPRITES	179
TOUS LES RUBANS PAR 3	144
DIGITALISEUR VIDI ST	990

DISQUETTES

Réf.	10	20	50	100
3" 1/2 DFDD	89 F	150 F	360 F	690 F
5" 1/4 DFDD	60 F	110 F	260 F	480 F

** Disquettes certifiées 100% garantie 5 ans.*
Livrées avec enveloppes + étiquettes Neutres (Verbatim)

BOITIER PLASTIQUE 3.50"...... 30 F Les dix

NOUS LIVRONS TOUJOURS LES DERNIERES VERSIONS

BON DE COMMANDE *EXPRESS* à retourner à **JESSICO** - B.P 693 - 06012 NICE CEDEX

EXPEDITION GARANTIE PAR COLISSIMO
GAGNEZ DU TEMPS ! Commandez par 93.51.61.30 **PAR MINITEL 3615 CODE JESSICO - OUVERT 7 JOURS SUR 7**

TITRES (garantie echange immediat)	Qte	Prix	Montant
		S/ TOTAL	
		PORT	25
		TOTAL	

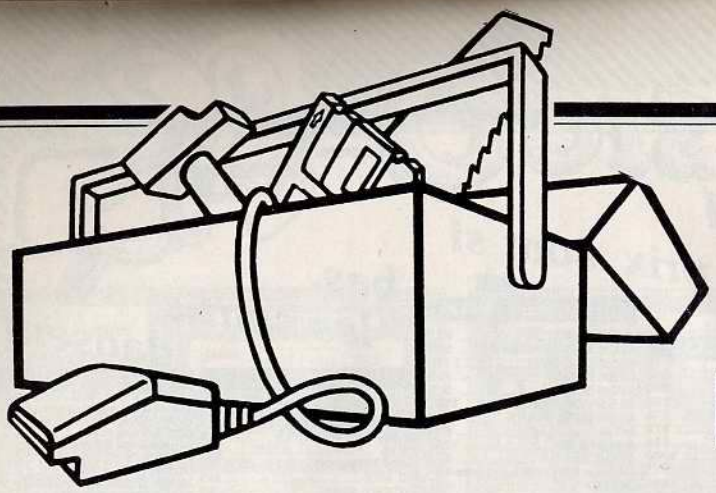
PORT LOGICIEL JEUX **18 F**
DOM TOM + ETRANGER **50 F**

Je joins un cheque ou mandat-lettre
Je paye par carte bleu et je complete les 2 lignes ci-dessous
carte bleue

NOM
PRENOM
N° ET RUE
VILLE
CODE POSTAL
SIGNATURE OBLIGATOIRE

precisez votre ordinateur DISC ... K7 ...

BOUTIQUE A NICE HOLLYWOOD STAR 8 BD. JOSEPH GARNIER



TWIST 2

Vous ne savez pas Twister ? Rien de plus simple. Après avoir chargé successivement deux (ou plusieurs) programmes en mémoire dans votre ST, passez de l'un à l'autre dans le temps qu'il faut pour appuyer simultanément sur les 3 touches Shift-G, Shift-D et Control. Les utilisateurs de Twist savaient déjà, et avec Twist 2, édité par la société Upgrade, c'est la même chose sauf que c'est mieux.

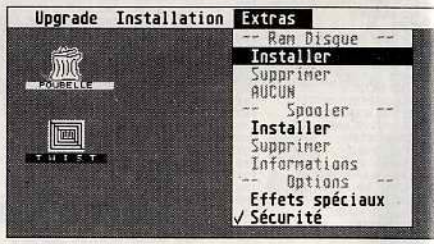
Il n'y a pas effectivement de différence notable de fonctionnement entre Twist et Twist 2. Ce dernier offre désormais la possibilité de travailler avec plusieurs accessoires (Twist n'en autorisait qu'un seul), et il met à votre disposition un spooler d'imprimante. Il a par contre simplifié et amélioré les opérations d'allocation de mémoire, étendu son domaine de compatibilité, amélioré sa fonction ram-disque et permet de charger jusqu'à 16 applications simultanées, en fonction de votre mémoire vive disponible.

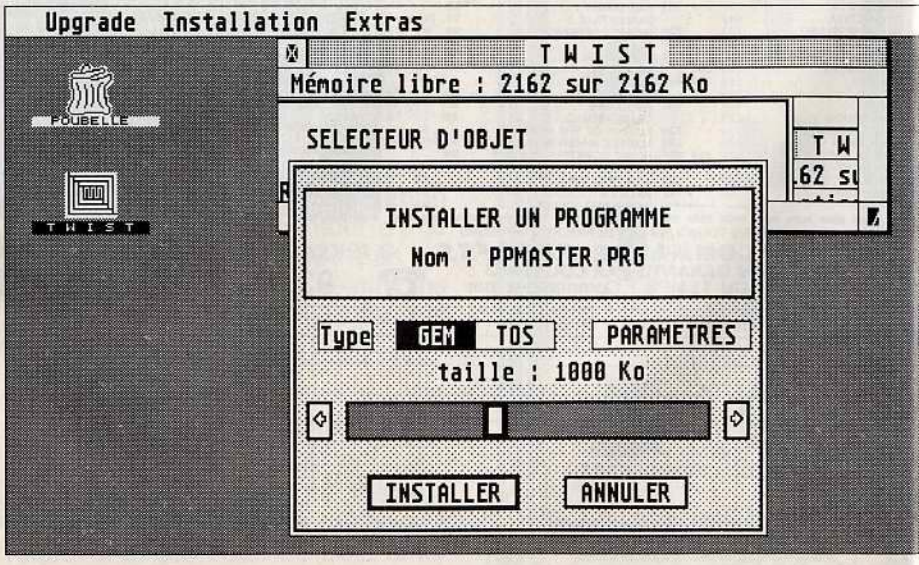
Par sa fiabilité, Twist s'était rapidement imposé comme le principal switcher non-dédié sur ST. Faire cohabiter plusieurs logiciels dans la mémoire, implique une rigueur de programmation qu'il assumait avec le plus grand sérieux, s'attachant à limiter le plus possible les dégâts en cas de lâchage d'une des applications en activité. Il permettait donc de bénéficier de toute la puissance de travail qu'offre la communication instantanée entre deux programmes, surtout lorsque celle-ci est renforcée par la rapidité de sauvegarde d'un disque-ram.

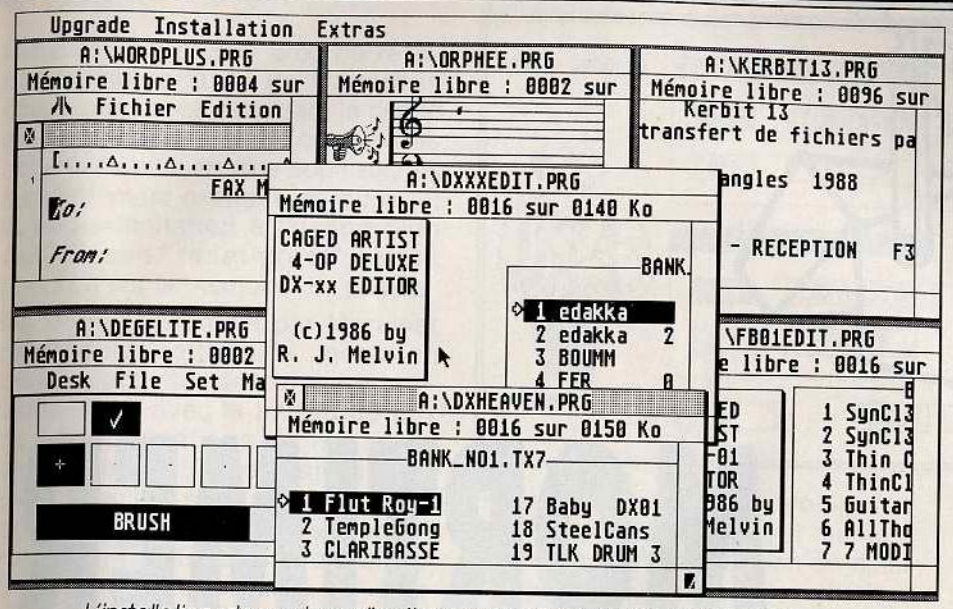
VIVE LA MÉMOIRE VIVE !

Twist se présente au lancement sous deux variantes : Twist 2 qui continue à n'accepter qu'un seul accessoire, et TwistAc plus généreux, mais au prix d'une réplication en mémoire de ces accessoires autant de fois qu'il y aura de programmes installés. Pour des raisons de même nature, à savoir l'espace physique alloué à chaque logiciel, si vous installez un disque-ram après avoir lancé une application, celle-ci ne le reconnaîtra pas. Se présente donc d'emblée la question à maîtriser d'urgence : la gestion de l mémoire.

La nouvelle version facilite le choses en fournissant avant lance ment une estimation plus précis des exigences du programme qu l'on se propose de charger. L fenêtre qui affiche chaque program me mentionne de plus l'utilisatio qu'a fait celui-ci de la mémoire qu lui était proposée. Mais seule évi demment l'expérimentation vou permettra d'arriver à optimiser vér tablement vos allocations mémoire Notez que certains programme supportent mal la limitation de leu







L'installation de sept applications, avec 2.5 mégas de mémoire

espace de travail, il vaut donc mieux commencer par pratiquer une certaine générosité, et ne pas hésiter dans un premier temps à sauver, avant d'activer les options avancées des applications installées. Twist 2 a aussi amélioré son estimation de la mémoire totale disponible en y incluant la part qu'il occupe lui-même. Les configurations sont toujours sauvegardables sous le nom de liasses. Elles correspondent surtout à une utilisation avec un disque dur, et sont rechargées automatiquement à chaque utilisation de la machine, c'est bien pratique.

LE RAM-DISQUE

C'est, nous l'avons déjà dit, un complément très logique de ce type de programmes. Il convient ici de l'installer préalablement aux logiciels qui doivent y avoir accès. Il peut également être désinstallé en cours de séance. Le ram-disque de Twist dispose d'une sécurité bienvenue sur les facéties d'une application, qui permet de retrouver les dernières sauvegardes. Mais une protection contre les pannes plus graves serait aussi la bienvenue. Certains ram-disques la garantissent... et on leur en sait parfois grâce. Le ram-disque vous entraînera lui aussi dans une arithmétique des octets. Pensez qu'il a surtout une vocation de buffer et de sécurité, fichier par fichier, et qu'il n'a pas besoin d'être énorme, mais souvenez-vous qu'il vaut souvent mieux un petit disque-ram que pas du tout.

SPOOLER

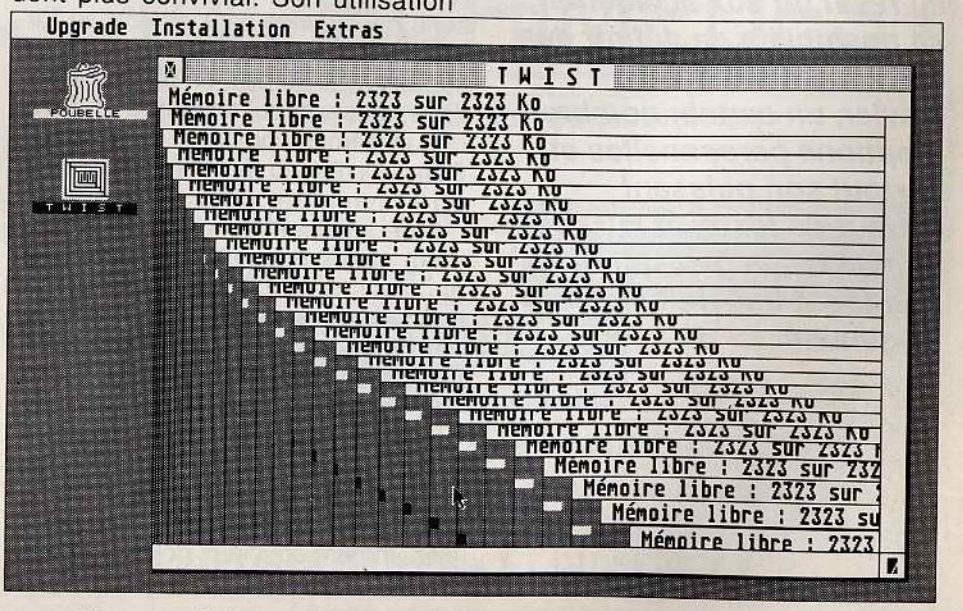
Installable à partir du bureau Twist il prend tout son intérêt lorsque l'or travaille avec des programmes comme Publishing Partner Master et UltraScript du même éditeur (Upgrade), puisqu'il permettra de poursuivre l'impression d'un fichier Publishing Master tout en continuant à Twister entre ces deux applications. Pour l'impression avec la SLM 804, il faudra penser à réserver environ 1 Méga d'espace de calcul pour cette dernière, qui, on le sait, utilise la mémoire de l'unité centrale.

DONC...

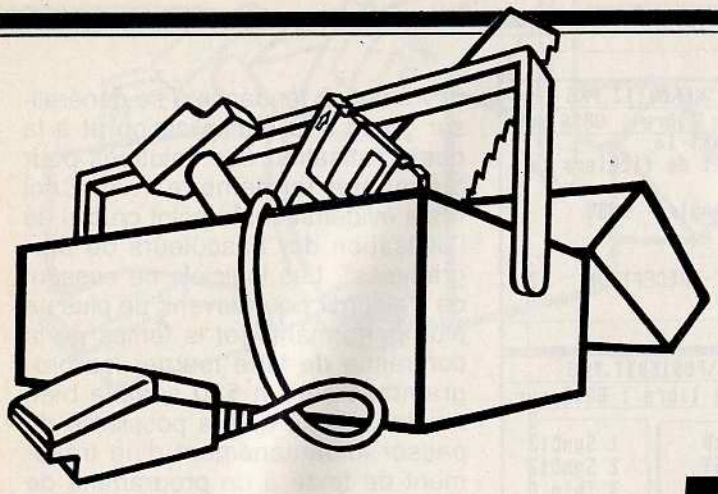
Twist qui était déjà un programme éminemment recommandable a bénéficié d'améliorations qui le rendent plus convivial. Son utilisation devrait avoir tendance à se généraliser grâce à la multiplication et à la démocratisation des solutions pour augmenter la mémoire vive. Ceci reste évidemment le point crucial de l'utilisation de "basculeurs de programmes". Les logiciels ne cessent de s'alourdir pour devenir de plus en plus performants, et le temps de la contrainte de faire tourner les programmes sur un 520 semble bien révolue. Reste que la possibilité de passer instantanément d'un traitement de texte à un programme de PAO ou de dessin, ou d'un séquenceur musical à un éditeur de son de synthétiseur offre des avantages à ne pas négliger. De très nombreuses applications sont envisageables. Twist m'avait permis par exemple sous son ancienne forme, de récupérer en un temps record toute une bibliothèque de sons, à l'occasion d'un changement d'éditeur de synthétiseur (le nouveau ne reconnaissant pas les banques de l'ancien). Cette opération aurait demandé des heures fastidieuses sans cette possibilité de bascule instantanée entre les deux logiciels.

Twist m'avait déjà impressionné dans sa première version par sa capacité à accepter un programme pas du tout GEM, dont l'affichage se trouvait certes un peu perturbé, mais qui acceptait de fonctionner. Tout concourt donc à garantir la fiabilité essentielle à ce type de programme.

François Auboux



Non, ce n'est pas un bug, mais Twist qui regarde Twist qui regarde Twist...



MORTIMER

Ram-disque, turbo-gestion des accès disques... durs, spooler, savantes calculatrices ? La liste des petits riens qui font tout pour transformer votre ST en secrétaire stylé s'allonge régulièrement. Pour espérer s'imposer de nos jours, un programme d'utilitaires doit apporter quelques plus distinctifs. Mortimer possède bien les ingrédients qui sont devenus la norme du genre. Mais sa résistance au reset (et aux accidents), sa possibilité de définir des macrocommandes au clavier, un certain nombre d'options personnelles et surtout son puissant éditeur de texte, et la qualité de sa réalisation lui confèrent une indéniable originalité.

Les utilitaires sont des programmes qui occupent souhaitablement le moins de place possible en mémoire, que l'on installe préalablement à d'autres applications et qui permettent d'en améliorer ou enrichir certaines fonctions, comme les accès disques, ou de bénéficier de mini programmes accessibles sans quitter l'application en cours. Ils se présentent soit sous forme d'accessoires de bureau, ils ne marcheront alors qu'avec les applications qui respectent le GEM, soit sous forme de programmes que l'on lance avant les autres et que l'on peut en général installer dans un dossier 'Auto', pour les appeler quand besoin est.

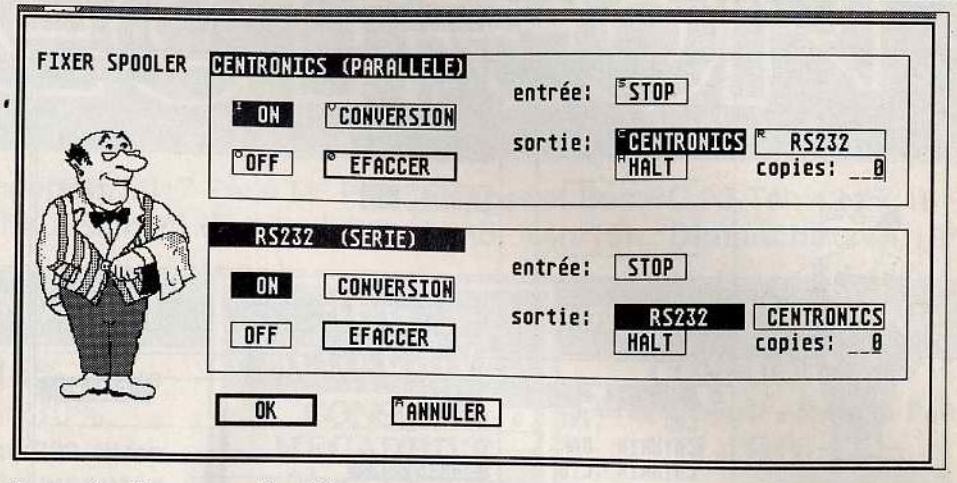
La concurrence sur ce marché est rude. Les programmes que nous utilisons quotidiennement offrent, au fur et à mesure de leurs mises à jour successives, de plus en plus de fonctions annexes, qui tendent à en faire des systèmes complets, autonomes. Dans le créneau des utilitaires, il faut être imaginatif pour arriver à se rendre indispensable. Il vaut également mieux présenter de sérieuses garanties de fiabilité car les réputations sont tenaces et plus faciles à perdre qu'à gagner. Or certaines propriétés spectaculaires des softs les plus récents, les amènent à des flirts douteux avec des zones plus ou moins délicates de la mémoire et/ou du système, qui ne facilitent pas la tâche des développeurs d'utilitaires évidemment soucieux de la compatibilité de leurs produits.

La liste des services offertes par Mortimer est tout à fait intéressante. Elle correspond visiblement aussi bien aux besoins d'un utilisateur traditionnel d'applications courantes, qu'à ceux d'un adepte du basic Omikron, avec lequel il privilégie la communication. Les services essentiels de Mortimer comprennent un Ramdisque résistant au reset, un gestionnaire complet de fichiers, un spooler d'imprimante, une loupe active dans tous les programmes, une calculatrice évoluée, la possibilité de définir des macro-commandes de clavier différentes pour tout programme, une commande pour parquer le disque dur, et pour la bonne bouche un excellent éditeur de texte. Le programme optionnel Killer sera une sentinelle permanente de détection de virus, y compris dans leur variante Link.

RAM-DISQUE

Il y a diverses bonnes raisons d'utiliser un ram-disque dont les deux plus importantes sont la sécurité et la rapidité. Dans diverses circonstances, l'interruption de l'action créative que représente une sauvegarde peut nuire à sa qualité (musique, écriture), or le ram-disque est la plupart du temps spectaculairement plus rapide que le disque dur lui-même. Par ailleurs, en l'absence de disque dur (et même en sa présence), la résistance au reset et aux crashes les plus sauvages d'un ramdisque comme celui de Mortimer,

fait qu'on n'a plus le droit de retarder la sauvegarde du travail en cours, augmentant d'autant le risque de le perdre. Les programmes actuels de plus en plus sophistiqués sont rarement exempts de bugs, et la seule limitation à l'usage d'un ram-disque "reset-proof" est sa ponction sur la mémoire. La possibilité offerte justement par Mortimer de moduler cette taille, même en cours d'activité, se combine judicieusement avec la baisse du prix des augmentations de mémoire. Une taille réduite de ram-disque bien calculée, suffit dans de nombreux cas et permet d'éviter les regrets... tardifs. Des [Contrôle-S] d'un quart de seconde aussi souvent que nécessaire et il n'y a plus alors qu'à opérer une sauvegarde plus définitive, dure ou molle, lorsque la pression créatrice s'est atténuée. Voilà donc une première bonne utilisation de Mortimer.



ÉDITEUR DE TEXTE

C'est une des fonctions que j'attendais depuis longtemps sur le ST. Pouvoir disposer d'un bloc-notes ASCII, qui permette à tout moment sans quitter une application de noter des éléments récupérables ultérieurement dans un traitement de texte. L'éditeur de Mortimer est beaucoup plus qu'un bloc-notes et apparaît en fait comme un mini traitement de texte intégré. Il possède, entre autres, de puissantes fonctions de blocs, puisque ces derniers sont mémorisés même après des accès disque, ou après avoir quitté l'éditeur, ce qui permet de leur assigner un rôle dynamique. Il gère l'impression, dispose d'une fonction recherche/remplacement itérative. Quand je vous aurai dit qu'il permet de travailler sur quatre textes différents accessibles par les touches F1 à F4, qu'il permet d'éditer directement un fichier depuis le bureau GEM, vous conviendrez que pour un utilitaire... de ce prix, l'affaire est bonne.

GADGETS DIVERS

Rien de dépréciatif dans l'intitulé de cet inter destiné à regrouper un certain nombre de fonctions plus habi-

AGORA
L'INFORMATIQUE DU FUTUR

4 rue NOUVELLE
95290 L'ISLE-ADAM
TEL:(1).34.69.56.60

EXPEDITION DANS TOUTE LA FRANCE
VPC

32 BITS RISC = 7290 FF TTC !
LE MICRO NOUVELLE GENERATION
HAUTE RESOLUTION,256 COULEURS
HUIT VOIES STEREO,MULTITACHE...

AGORA... IMPORTATEUR EXCLUSIF DU
PROCESSEUR ARM3 30 MHZ /25 MIPS

L'ARCHIMEDES

DEMONSTRATION PERMANENTE !!!

CATALOGUE DE + DE 250 ARTICLES
PROGRAMMATION , PAO , DESSIN , MUSIQUE , RAY TRACING , VIDEO
EXTENSIONS MEMOIRE A300 et A3000
DISQUE DUR 45 ET 70 Mo,SCSI et ST506
CARTE SCSI A300 ET A3000
MONITEURS MULTIRESOLUTIONS
BIBLIOGRAPHIE DABBS COMPLETE...!

A 410+ 4 Mo RAM +HD 53 Mo +multisynchro .. 27 990 FF

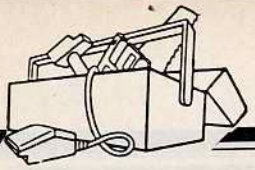
STOCK PERMANENT DE LOGICIELS

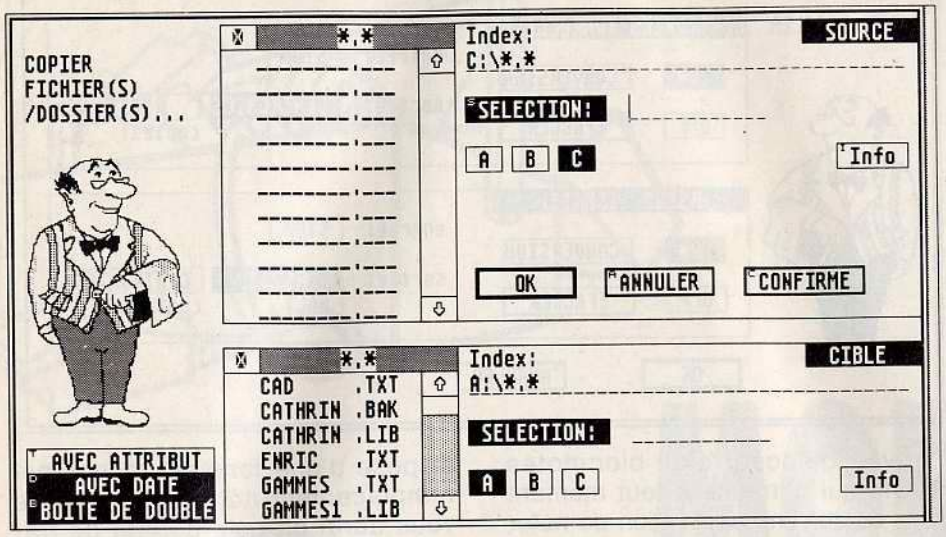
LE VIDC ENHANCER DISPONIBLE...
18 RESOLUTIONS SUPPLEMENTAIRES !

7 Mo de domaine public = 150 FF

version RISCOS SOLID RENDER, SOLID CAD,DELTA+ IMPRESSION,EUCLID,MOGUL,RISC BASIC,FORTH...

DES DEMOS , DU CHOIX ,DES IMPORTS,DES PRIX ET DU SERVICE...!





tuelles. La façon Mortimer de les gérer est au contraire tout à fait dans la ligne de ce que nous avons décrit précédemment et ne fait que confirmer le rôle d'auxiliaire sérieux qu'il prétend assumer.

Le spooler d'imprimante spoole, avec un bon luxe de paramétrage des sorties, dont la sortie nulle pour analyse des données. Mais il entretient de plus des relations cordiales avec le ram-disque qui leur permettent d'échanger de la mémoire si le besoin s'en fait sentir, ainsi qu'avec l'éditeur de texte pour lequel il veut bien accepter de s'interrompre afin de "swapper" des fichiers, que de courtoisie!
La gestion des entrées et sorties disques est aussi sophistiquée que faire se peut. Elle inclut entre autres l'effacement de dossier, mais surtout une fonction Copie de fichier hyperformante puisqu'elle permet de gérer tous leurs attributs.

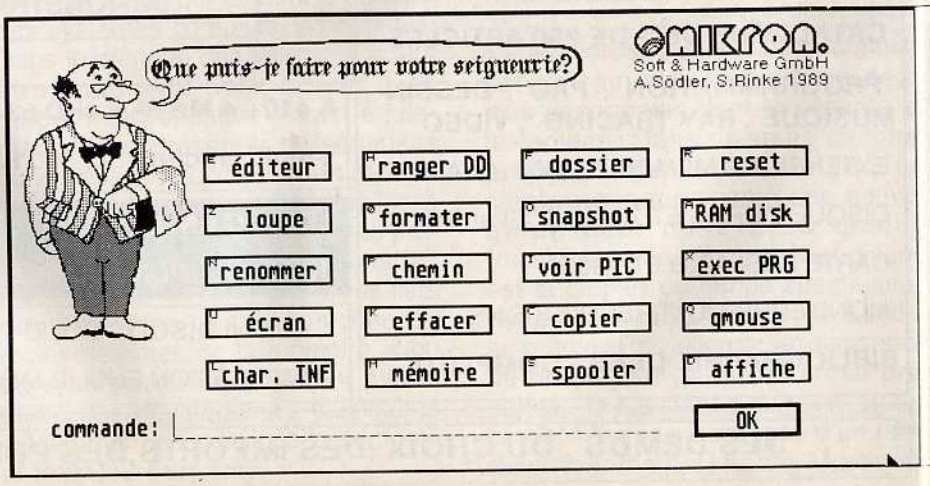
L'élaboration des macro-commandes a été tout aussi bien pensée. Mortimer offre en effet la particularité d'être modulaire, la description de sa structure étant accessible et modifiable à tout moment par appel d'un fichier INF. Il est par conséquent possible lors d'un changement de programme, de faire appel à une nouvelle configuration de ces macro-commandes, donc d'en avoir autant que de programmes que l'on utilise. C'est en fait même la possibilité de changer toute la configuration de Mortimer qui est offerte par cette option, donc d'en sauver une bibliothèque. Ces commandes s'entendent par ailleurs dans un sens très large, car elles peuvent par exemple permettre par passage en ASCII et assignation de blocs, des récupérations de base de données ou d'agenda entre des programmes à fichiers incompatibles.

La calculatrice est suffisamment polyvalente pour être honnête. Fonctions mathématiques, reconnaissance de la notation scientifique, niveau de parenthèses, conversions et calculs directs en binaire, en hexadécimal ou en octal, avec une option (liée aux macro-commandes) de récupération des résultats dans l'application en cours. Le Snapshot (capture d'écran sous forme de fichier graphique) connaît les formats PIC, Degas et IMG. La souris est paramétrable grâce à un polynôme du troisième degré (oh la vilaine!). Cela veut quand même dire que l'on peut gérer son déplacement, sa vitesse et son accélération (cf. Chap. Dérivées de votre livre de maths).
La loupe... grossit (x4), en maintenant autant que possible les fonctions des programmes, et le parqueur du disque dur... devinez ! A noter également une simulation du reboot à froid ou à chaud avec la possibilité de désactiver le ram-disque, ou Mortimer, ou les deux, ou de l'envoyer chercher la mousse à bronzer.

GÉNÉRIQUE DE FIN

Je vous laisse donc imaginer ma détresse, lorsqu'après ce long dithyrambe je vous confierai que Mortimer inhibe le fonctionnement du programme qui remplit le plus habituellement mon ST : CuBase ! Je n'incrimine personne car si j'utilise CuBase c'est notamment pour la surpuissance que lui confère son M_ROS, noyau "multitâche" alors que le ST ne l'est pas. Or, comme signalé dans l'introduction, ce genre d'outil plonge à des niveaux d'intimité du ST où les plus bas sentiments prévalent, conflits et jalousies menaçant. Ainsi Rédacteur 1 et 3 acceptent de tourner, mais c'est Mortimer cette fois qui se laisse impressionner. Les susceptibilités d'assembleur sont au rendez-vous. Ne vous laissez pas intimider, ce serait dommage (mais vérifiez quand même si vous avez un programme favori), ce sont les seules incompatibilités que j'ai rencontrées, et que j'aimerais bien voir contournées, mais à qui la "faute" ?... D'autant que, je me répète, Mortimer possède bien le savoir-faire et la discrétion de ces serviteurs anglais auquel il emprunte son image, et que pour les quelques 64 K d'assembleur qu'il prélève sur la mémoire, il en fait vraiment des tonnes.

François Auboux



ELECTRON

12 Place de la Porte de Champerret 75017 Paris M° Pte Champerret Bus PC,92 Tél: 42 27 16 00
Ouvert 7 jours sur 7: Mardi au Samedi de 10h à 20h, Lundi 14h/19h, **Dimanche 14h/18h**

ATARI ST

520 STE +
Moniteur Couleur + Joy + Maupiti ou Tennis Cup
5490 Frs

1040 STE +
Moniteur Couleur + Maupiti + Tennis Cup + Joy
6490 Frs

MEGA ST1 +
Moniteur SM 124 Mono + 50 Disquettes vierges
5990 Frs

Lecteur ST/AMIGA externe 720K DF
720 Frs

ATARI XE

130 XE Peritel	590
Lect K7 XL/XE	250
Console XE game	490
Kit Clavier + K7 XE	390
Imprimante A1027	490

ATARI LYNX

cable de liaison + (4Jeux)
CALIFORNIA GAMES
Adaptateur Secteur
1490 F

ELECTROCOP	NC
BLUE LIGHTNING	NC
GATE OF ZENDECON	NC
CHIP'S CHALLENGE	NC
GAUNTLET	NC

NOUVEAUTES
47 66 11 77

SOFT ST

15% News
30%
sur 200 titres
Nouveautés
Tous les jours

GAMEBOY

+BATMAN
890 F

40 titres disponibles
1 Jeu: 270F
500F les 2

ATARI PORTFOLIO
+ Interface Paralléle
2290 F

MUSIQUE ST

PRO 24 3,0	2630
PRO 12	680
CREATOR	2350
NOTATOR	3900
FM MELODY	790
CUBASE	4150
STUDIO 24	1120
TRACK 24	580
PRO SCORE	1530
ST REPLAY PRO	1290
MIDIMIX	89

DEMONSTRATION SUR RV

Imprimantes

STAR LC 10
1890 F
STAR LC 24/10
2990 F

---SEGA---

MEGADRIVE

CONSOLE MEGADRIVE
+ 2éme Manette
+ OSAMATSUKU
1790 F

ADAPTATEUR SEGA 8 BIT DISPONIBLE

NOUVEAUTES
DJ BOY, WIP RUSH
ASSAULT SUIT
THUNDER FORCE III
BASE BALL
SUPER SHINOBI, GOLF
MONACO GRAND PRIX
FINAL BLOW
MAJOR COP RYU
WORLD CUP SOCCER
GOLDEN AXE
KEN LAST BATTLE
390 F pièce

SOFT AMIGA

INCROYABLE!!!
2 Softs achetés
1 Gratuit

SNK NEO GEO

CONSOLE	3950
BASE BALL	1950
NAM 75	1950
MAGICIAN LORD	1950
GOLF	2250

LOCATION
VENTE REPRISE
NOUVEAUTES
47 66 11 77

NEC PC ENGINE

CONSOLE NEC
+ Dragon Spirit + Monster Path
1490 F

CD NEC à 450 F pièce
DEATH BRINGER
MILITARY COMMANDO
SHANGAI II
SUPER DARIUS

Jeux NEC à 390 F pièce
Barumba, Formation Soccer, Splatter House, Blue Blink, Image Fight, Batman, Be Ball

Tous les jeux NEC disponibles à des prix fous

SUPERGRAFX
+ BATTLE ACE
2490 Frs
GRAND ZORT 390 F
Ghouls & Ghost, The Strider NC

COMMODORE AMIGA

Cadeau ELECTRON
50 Disquettes pour l'achat d'un AMIGA, 100 Disquettes pour l'achat d'un AMIGA Couleur

AMIGA 500 Monit 1084 S + Cadeau
5690 F

AMIGA 500 + Cadeau
3490 F

50 Disquettes KONICA
250 F

100 Disquettes KONICA
480 F

10 Disquettes KONICA
La boîte
80 F

500 Disquettes KONICA
420 F/ 100

OFFRES VALABLES DANS LA LIMITE DES STOCKS DISPONIBLES

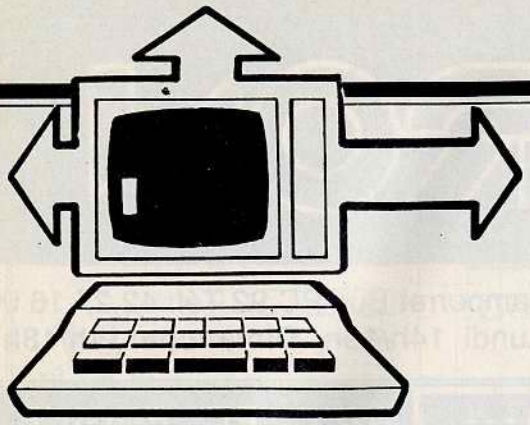
NOM.....PRENOM.....
ADRESSE.....
C.P. .....VILLE.....
CHEQUE MANDAT CARTE BLEUE
N° |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| DATE EXP.|_|_|_|_|

PROMOS ET NOUVEAUTES:
TAPEZ 3615 ELECTRON

Je vous passe commande de l'offre suivante:

Désignation	Qté	Px Unit	Prix Tot

Port Disquettes 40 F les 50, 55 F les 100, 100 F au delà ---------- PORT
Port Accessoire : 50 F Port Matériel : 150 F ----------
Signature Parents si mineur: TOTAL ----------F TTC



GESBARRE

Vous vous souvenez certainement de Gestocks, ce logiciel de gestion de stocks, clients, fournisseurs et de facturation que nous avions testé dans notre numéro 37, remarquable pour sa simplicité et sa facilité d'utilisation "malgré" sa puissance. Voici aujourd'hui "Gesbarre", du même auteur et toujours édité par Arobace, qui n'est rien d'autre qu'un éditeur de codes à barres, aussi remarquablement simple que son "grand frère" et remplissant parfaitement le rôle qui lui est assigné comme nous allons le voir.

SIMPLICITÉ

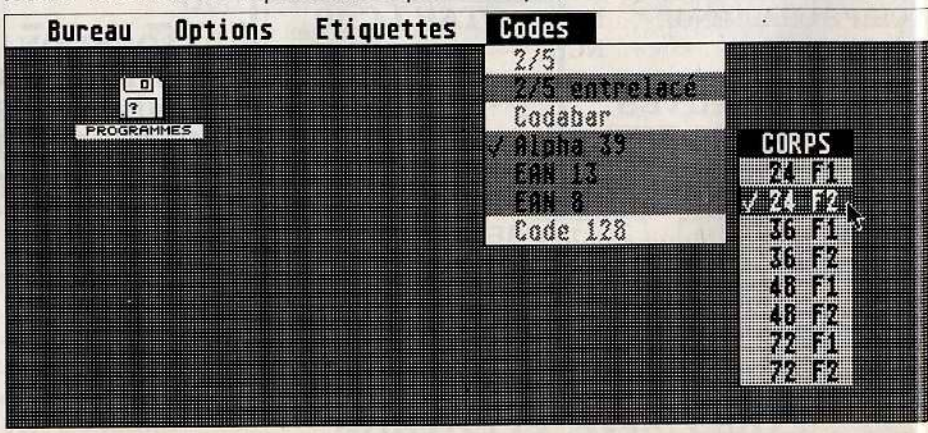
C'est sans doute le terme qui convient le mieux pour décrire l'ensemble des manipulations relatives à Gesbarre. Pour l'installation, avant tout, le logiciel prend tout en charge grâce à un module spécifique, que ce soit sur disquettes ou disque dur, en créant les répertoires adéquats et en rangeant tous les fichiers sans que l'utilisateur n'ait à se poser de question. Étant donné que pour les sorties graphiques, Gesbarre utilise GDOS (dont la "simplicité", par contre, n'est pas remarquable...), il y aura création automatique d'un fichier ASSIGN.SYS pour les fontes utilisées. Mais luxe suprême, si l'utilisateur dispose déjà d'un tel fichier (qu'il a mis en général quelques heures à comprendre et à constituer au fur et à mesure d'expériences plus ou moins traumatisantes), Gesbarre ne "l'écrase" pas et crée un fichier parallèle, dont la gestion ne pose aucun problème avec G+Plus. En effet, Gesbarre est automatiquement livré avec G+Plus, ce superbe utilitaire lui aussi édité par Arobace, qui permet de choisir un fichier Assign.Sys en fonction de chaque application et en spécifiant son chemin d'accès. Un confort total, donc, la configuration générale du programme ne posant aucun problème. La documentation, d'une vingtaine de pages, est limpide et surtout très didactique vis à vis de GDOS et ses fontes, avec en plus quelques informations intéressantes sur les codes à barres en général.

LES CODES A BARRES

Sans doute n'est-il pas inutile de revenir quelques instants sur ces signes tangibles de notre société de consommation, et que tout un chacun connaît désormais pour peu qu'il soit allé une fois dans sa vie acheter un bien quasi-quelconque dans un commerce... C'est en effet un procédé de plus en plus répandu et répondant à des normes internationales, permettant d'identifier rapidement, à l'aide d'un lecteur optique, les références d'un produit, sa famille, son type, son numéro de série, etc., toutes informations représentées par des petites barres à l'épaisseur variable, sous lesquelles figurent éventuellement des chiffres et/ou des lettres. On en trouve plusieurs types standard, et Gesbarre les propose aussi, au nombre de sept : les "2/5" normal et entrelacé, les "EAN 8" et "EAN 13" (les plus universels), le "Codabar" (employé principalement en pharmacie), l'Alpha39 et le Code128 (le plus complet, un "standard" industriel). Les cinq premiers ne contiennent que des infos numériques tandis que les deux derniers sont alphanumériques (lettres plus chiffres). Lors de l'installation, Gesbarre demande les types de codes à installer (si l'on ne désire pas les utiliser tous) afin de réduire la place prise par les fontes GDOS, très gourmandes en place mémoire. La boîte d'installation demande aussi le type d'imprimante utilisée (pour le choix d'un driver spécifique), en proposant les 9 et 24 aiguilles compatibles Epson ainsi que la SLM804 (laser Atari) mais avec cette dernière, un Mega ST2 sera indispensable, du fait de l'absence de mémoire dans l'imprimante qui oblige à "composer" la page dans la mémoire de l'unité centrale.

UTILISATION

Le logiciel charge une page écran simple, avec des menus "diététiques"

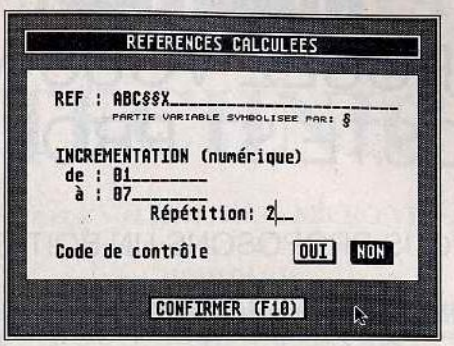


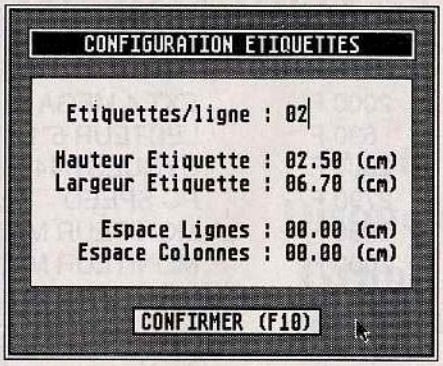
Une partie de l'écran principal

(peu chargés, chacune des options possédant son raccourci-clavier facile à mémoriser), dont un "hiérarchique" s'il vous plaît. Une icône ("Programmes") attire immédiatement l'attention, c'est le lancement possible d'une autre application, un programme extérieur quelconque comme un éditeur de texte, une base de données, ou Gestocks lui-même, avec lequel Gesbarre est évidemment interfacé afin d'utiliser ou de compléter la base "Produits et Références". L'opération effectuée, on revient automatiquement à Gesbarre, c'est pratique. L'écran principal présente aussi une ligne pour l'heure et la date du système, sur laquelle il suffit de cliquer pour modifier son paramétrage.

L'utilisation générale de Gesbarre se résume ensuite à trois opérations principales : l'édition du code, la préparation de l'impression et son paramétrage (format de page et d'étiquettes) et l'impression elle-même. Le principe consiste tout d'abord à choisir le type de code à éditer et la taille des caractères, puis de préparer le format de la page selon 6 options : sa hauteur, sa largeur (deux paramètres imposés par GDOS en fonction du driver utilisé), ses offsets en X et Y, sa hauteur et sa largeur (zone d'impression). Tout cela permet de préparer précisément sa page, et si vous n'êtes pas sûr d'obtenir le résultat escompté, il sera toujours possible de la visualiser avant de l'imprimer. Il faut aussi préparer les "étiquettes", en choisissant le nombre d'étiquettes par ligne, leurs hauteurs et largeurs, et les espacements entre les lignes et les colonnes d'étiquettes. Là aussi, le mode pré-visualisation permet de s'assurer de la justesse des paramètres saisis, et notamment de constater si le code choisi vient correctement s'insérer dans l'étiquette déterminée.

L'étape suivante consiste à paramétrer les codes eux-mêmes pour les impressions en série, à partir soit d'une référence unique, soit d'une référence calculée. La référence unique revient à éditer n fois un même code déterminé, dans lequel, nous ne l'avions pas précisé, il est possible de rentrer un code de contrôle, et qui n'est autre que le fameux "checksum" bien connu des programmeurs. La référence calculée, elle, permet d'éditer une suite





de codes à incrémentation automatique, par exemple mille étiquettes allant de la référence 0001 à 1000, en rentrant un ou plusieurs jokers (§) pour la partie variable.

Tout cela se fait très simplement et rapidement, mais le plus spectaculaire réside dans la possibilité d'importer sous Gesbarre des références déjà contenues dans une base de données, un tableur ou même un fichier texte. En respectant le formatage des champs de saisie, il est alors possible d'utiliser des séries complètes de références, de façon complètement automatique. Cela peut être très intéressant lors d'une utilisation de références disparates ou non successives.

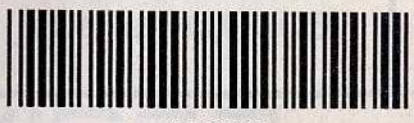
Une dernière chose importante : les paramétrages des formats de page et d'étiquettes sont sauvegardables séparément, et l'utilisateur peut ainsi se constituer une bibliothèque de fichiers de paramètres divers, dont un sera chargé par défaut au lancement du logiciel.

VOILÀ...

...pour ce rapide "tour du propriétaire", car tout est simple, facile et rapide au point qu'un enfant de 6 ans saurait parfaitement s'en servir. Particulièrement destiné aux gens peu habitués à "l'informatique", Gesbarre rendra d'inestimables services pour la gestion des stocks et des produits, en permettant de gérer soi-même et très proprement des codes à barres. Son prix semble raisonnable (environ 2500 F), compte tenu du fait qu'il est livré avec G+Plus et qu'il constitue typiquement ce que l'on nomme une "application verticale". Enfin, pour terminer, lançons un petit coup de chapeau à l'auteur, qui prouve une fois de plus que ergonomie, simplicité et facilité d'utilisation, en programmation, font partie intégrante d'un cahier des charges qui se respecte...

Laurent Debarre

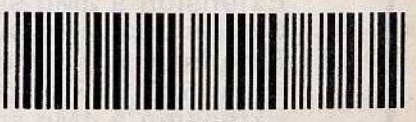
ABC01X


ABC01X

ABC02X

ABC02X

ABC03X


ABC03X

Impression, sous SLM 804, d'une série de codes "Alpha39" en référence calculée

NOUVEAU

COMPOSEZ VOUS MEME VOTRE POSTE ST PROFESSIONEL

NOUS VOUS PROPOSONS UN BOITIER AVEC :

1 CARTE MEGA ST, 1 ALIM 200W, 1 LECTEUR 720 K, 1 CLAVIER MEGA : **5990 F**

VOUS CHOISISSEZ VOS OPTIONS :

EXT 2 MEGA	2000 F	EXT 4 MEGA	3500 F
2 ème LECTEUR 3"1/2	690 F	LECTEUR 5"1/4	690 F
DISQUE DUR 20 MEGA + CARTE SCSI	2590 F	SYQUEST 44 MEGA + SCSI ET CARTOUCHE	5490 F
CARTE 16 MHZ	2790 F	PC SPEED	2690 F
SPECTRE GCR	3990 F	MONITEUR MONOCHROME	990 F
MONITEUR COULEUR	2000 F	MONITEUR MULTISYNC	5990 F

RENSEIGNEZ VOUS ...

Le STACY EST ICI
venez le voir
1 Mo RAM
20 Mo Disque dur

NC

avec Spectre
(émule Macintosh* portable)

NC

PAO / CAO / DAO / .....

METTEZ UN TIGRE DANS VOTRE MOTEUR !

ATARI MEGA ST
16 Mhz
4 mégas de mémoire

11990 F

ATARI PC2 HD

Le compatible complet :

disk 5"1/4
Disque Dur 30 Mo
CGA / EGA
HERCULES / MDA

6490 F. TTC

NOUVEAUX PRIX ! KITS EXTENSION MEMOIRE

Quelque soit votre système, nous avons une solution.

Extension à 1Mo sur STF	690 F
Extension à 1Mo sur STE	490 F
Extension à 2Mo sur STE	1490 F
Extension à 2Mo sur Mega ST1	1990 F
Extension à 2,5 Mo sur 260, 520 ST ou 520 STF ou 1040 STF	2290 F
Extension à 4Mo sur Mega ST2	1990 F
Extension à 4Mo sur Mega ST1	3490 F
Extension à 4Mo sur STE	2990 F

Certaines extensions nécessitent de bonnes connaissances de montage electronique. Pose possible par nos services sous 8 heures. Consultez nous.

LES SCANNERS

CANON IX12F
scanner à plat 300 DPI
12990 F TTC

HANDY SCANNER TYPE 10
400 DPI105 mm de large
2290 F TTC

* Macintosh est une marque déposée d'Apple Inc.

Pour compléter son équipe technique

MICRO VIDEO PARIS

recherche

UN TECHNICIEN

envoyer CV et prétentions à MICRO VIDEO
8 rue de VALENCIENNES
75010 PARIS

LES MV-KITS

Disques durs S.C.S.I.

KIT 1 :
Disque dur 20Mo + carte
2690 F

KIT 2 :
Disque dur amovible + carte
+ cartouche 44 Mo
5490 F

KIT 3 :
Disque dur amovible + carte
+ cartouche 44 Mo, boitier, alim

MICRO VIDEO se lance sur le Macintosh *

avec des prix Atari !

carte SE 2Mo
Disque dur 20 Mo
Grand Ecran A4

idéal pour WORD 4 / XPRESS 3.0

19990 F Hors Taxe

GRATUIT

jusqu'au 30 Août 90
1 Mo
au prix de 500K

Pour tout achat d'un 520 STE, nous vous offrons l'extension 500K supplémentaire.
ce qui vous donne un 1024 STE pour
3490 F

MECANIQUES DISQUE DUR

20 Mo SCSI
1790 F TTC
SYQUEST 44 Mo
4290 F TTC
Carte DMA/SCSI

Ces prix sont valables jusqu'au 30 Août 90 et annulent les précédents. Opérations dans la limite des stocks disponibles Certains articles et certains prix peuvent être différents en Belgique.! RENSEIGNEZ VOUS DES DELAIS POUR LA PROVINCE

MICRO VIDEO

la passion d'un spécialiste, la puissance d'une chaine.

PARIS
8, rue de Valenciennes 75010 Paris
☎ 40.37.92.75 / 40.34.97.80 +
Ouvert du Mardi au Samedi de 10H à 19 H
Métro: Gare de l'Est / Gare du Nord

BORDEAUX
3, cours Alsace et Lorraine
33000 Bordeaux

TOURS
81, rue Michelet
37000 Tours
☎ 47.05.78.50

DAX
56, Avenue Victor Hugo
40100 Dax
☎ 58.74.18.63

NANCY
55, rue des 4 églises
54000 Nancy
☎ 83.37.06.47

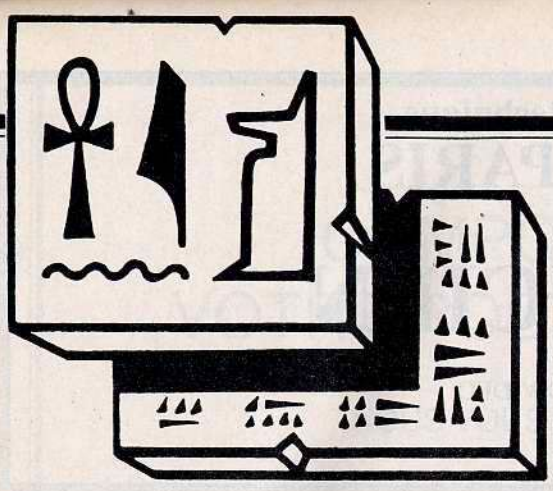
PERPIGNAN
8, avenue de Grande Bretagne
66000 Perpignan
☎ 68 34 24 40

LYON
11,12 cours Aristide Briand
69300 Caluire

BELGIQUE

BRUXELLES
1, rue Dons
1050 Bruxelles
☎ 02 / 648 9074

DINANT
21 place communale
5198 ANHEE
☎ 082 / 611451



GfA-GUP

GfA-GUP est ce nouvel utilitaire dont nous vous avions parlé lors de notre compte-rendu sur le CeBit à Hanovre, et qui vient d'être importé et francisé par les Editions Micro-Application. C'est en fait une bibliothèque de routines GfA qui simplifient la programmation de l'interface utilisateur. Le GfA tente ainsi de rattraper l'Omikron et son Easy-GEM...

Tout d'abord, il faut vous dire que dans ce test, nous n'avons pas grand-chose à vous montrer à part des pages de GfA... En effet, GfA GUP est un gros fichier en GfA 3.0 de 54 Ko contenant un nombre certain de routines (certaines sont "Private") dont 61 sont accessibles à l'utilisateur. Quel est le but de GfA GUP ? Ceux qui ont déjà développé sous Unix et la Toolkit de X-Windows savent ce qu'est une Widget : c'est une structure décrivant une fenêtre ou un objet ainsi que son contenu. Dans un tel système, ouvrir une fenêtre 50x50 sur une bit-map 1000x1000 avec un scrolling automatique non géré par l'utilisateur est facile... Ça, c'est le but à atteindre, GEM ne faisant que transmettre ses événements à l'utilisateur en personne qui se charge de la gestion de l'interface.

Comme on fait de la programmation, commençons par le principal : la doc... J'ai pas mal souffert en la lisant, si, si... Imaginez le manuel du Laser C sans une ligne d'exemple et vous obtiendrez à peu près ce que j'ai sous les yeux en ce moment : dur ! Finalement le sempiternel "Description - Parameters - Synopsis" a du bon. Enfin, au bout d'une exploration complète de ses 80 pages, voilà ce qu'il faut savoir : GfA GUP "peut" être considéré comme une couche supérieure à GEM en ce sens qu'il récupère la quasi-intégralité des événements courants, gère tout seul les principaux, renvoie ceux qu'il ne "comprend" pas à l'utilisateur et apporte de nombreuses fonctions pratiques qui manquent au GEM standard. Mais attention, pas question de mélanger du bon GfA habituel avec du GfA GUP. Certaines fonctions du Basic deviennent interdites ou obsolètes.

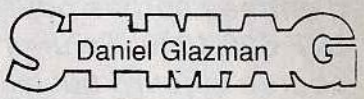
Exemple : vous voulez définir une fenêtre graphique contenant un cercle et une droite. Vous ouvrez la fenêtre en question et vous créez une PROCEDURE GRAPHIC(window,hor,ver) qui sera appelée à chaque fois qu'un événement REDRAW sera intercepté par GfA GUP... On est bien entendu encore loin des Widgets qui se redessinent toutes seules ! L'utilisateur doit toujours, en fonction du numéro de fenêtre transmis, tracer son graphique tout seul.

Côté programme, c'est plutôt propre... Le code est assez condensé, et il n'est pas conseillé au néophyte de se plonger dedans pour comprendre pourquoi au moment 't' la procédure 'schmugle' récupère la liste des rectangles à redessiner. A noter que la version qui m'a été communiquée est faite pour les GfA 3.XX même si la doc la signale compatible avec le 2.02. On y trouve donc un bon millier de lignes très peu commentées (une version n'est pas commentée du tout pour prendre moins de place), séparées en "procédures non appelables" et "procédures appelables". Mais le grand désagrément, à mon sens, de GfA GUP est le suivant : dès que l'utilisateur veut récupérer des données concernant une fenêtre ou un objet, il travaille avec des variables globales (et il y en a un paquet !) organisées en tableaux, ce qui est d'une propreté et d'une rigueur folle... Celui (ou ceux) qui a programmé GUP devrait savoir que Get_Window_XY(w,VAR x,y), ça fait toujours plus lisible et plus propre que x=xcoor(1) y=ycoor(1)... Bref, il faut un sacré moment avant de pouvoir se passer de la documentation pour programmer cette petite Toolkit.

Je suis railleur, certes, mais les résultats ne sont quand même pas mal : un programme de démo est fourni (moi pas comprendre pourquoi avoir une version TOS 1.2 et une version TOS 1.4 ?) et j'avoue que cela tourne bien et vite. Fenêtres graphiques, fenêtres textes avec sélection simple ou multiple de lignes, bureau GEM et icônes, sliders et arbres d'objets, c'est impeccable et plutôt professionnel pour un programme de 265 lignes (plus 1100 de GfA GUP, bande de petits rigolos...).

La doc annonce que certains programmes commercialisés en ce moment utilisent GUP dans leur programmation. Il est certain qu'il y a un côté pratique à ne programmer que le strict minimum (qui peut encore être réduit) et qui peut ne pas être contrebalancé par un ralentissement obligé à l'exécution. Cela demande à mon avis tout de même un bel investissement de la part du développeur que de se plonger dans un code qui, n'ayons pas peur des mots, n'a pas l'air tout à fait finalisé... On aurait aimé que GfA GUP puisse attacher une bit-map ou un fichier texte à une structure de fenêtre et s'occuper lui-même des divers rafraîchissements et scrollings. Signaler, dans la doc, que la résolution actuelle peut se récupérer par un XBIOS et annoncer qu'en fonction de cette dernière, l'utilisateur devra faire un changement d'échelle fait un peu mesquin : cela demande deux lignes de programmation en plus pour l'auteur de GUP !

En conclusion, disons que GfA GUP est un produit intéressant à faible prix, qui séduira ceux que la programmation GEM pure et dure en GfA rebute, qui simplifiera aussi certains aspects de la programmation de logiciels "au look professionnel" mais qui obligera leurs auteurs à se détacher un peu de l'optique de GEM qui offre des procédures de rappel de paramètres. En fait, un code un peu jeune dont on attend qu'il vieillisse et s'améliore comme le bon vin, même s'il a déjà des qualités indéniables...



LES BONNES ADRESSES

ULTIMA
Place du Capitole
35 rue du Taur
31000 TOULOUSE
Tél. 62 27 04 37

ULTIMA
72-74 rue de Paris
59000 LILLE
Tél. 20 42 09 09
métro gare

MICRO VIDEO BELGIQUE
1, rue Dons
1050 Bruxelles
02 / 648.9074

26000 VALENCE

MICRO AVENIR
4, rue des Alpes
✆ 75. 55. 41. 19.

LE SPECIALISTE ST A VALENCE.

MICRO AVENIR
2, avenue de Romans
38500 Voiron
76.65.72.55.

MICRO VIDEO BORDEAUX
3, cours Alsace et Lorraine
33000 Bordeaux
56.79.34.89

MICRO VIDEO PERPIGNAN
8, Ave de Gde Bretagne
66000 Perpignan
68.34.24.40.

MICRO VIDEO LYON
11 cours Aristide Briand
69300 Caluire
72.27.14.74.

CARTE DE CONVERSION A/N-N/A 8 BITS

Voici quelques mois, nous vous avions proposé deux articles consacrés à la réalisation d'interfaces d'entrées/sorties pour le STF 520/1040. Celui-ci aborde aujourd'hui la réalisation d'une carte de conversion stéréo A/N-N/A 8 bits, vous permettant d'entrer dans le monde, ô combien étrange, de l'analogique...

LA CONVERSION A/N-N/A

Avant d'entrer dans le détail de cette carte, il est préférable de consacrer l'espace de quelques lignes à l'étude des circuits spécialisés, qui en constituent le coeur. La figure 1 vous présente le diagramme fonctionnel du convertisseur A/N 8 bits, AD 7575 de "Analog Devices", qui travaille selon la méthode des approximations sucessives. Celle-ci consiste à comparer, à l'aide d'un comparateur (COMP.) qui mérite bien son nom, la tension d'entrée (Vin) à une tension de référence, qui, en pas binaires successifs, se rapproche de la tension d'entrée. Cette tension de référence est obtenue à l'aide d'un convertisseur D/A commandé par un registre d'approximations successives (SAR). Cette conversion n'étant pas instantanée, il faut "mémoriser" la tension d'entrée pour éviter toute erreur de quantification durant la conversion.

Cette mémorisation s'effectue à l'aide d'un échantillonneur/ bloqueur (Track and Hold). La tension de référence Vref permet de déterminer l'étendue de mesure du convertisseur A/N qui est comprise entre 0 volt et Vrefx2. Une fois la conversion terminée, une valeur binaire proportionnelle à la tension échantillonnée, comprise entre $00 et $FF, est fournie sur les bits DB0 à DB7. La relation entre la tension d'entrée (Vin) et la valeur binaire (Data) obtenue est :

Data(LSB's)=(256*Vin)/(2*Vref)+0.5.

Donc si Vref est égale à 1.23 volt, l'étendue de mesure du convertisseur A/N sera comprise entre 0 et 2.46 volts et les valeurs binaires respectives obtenues sont $00 et $FF. Ce circuit comporte plusieurs entrées de commandes qui permettent, à l'aide d'une logique de contrôle, de synchroniser les différentes étapes d'une conversion.

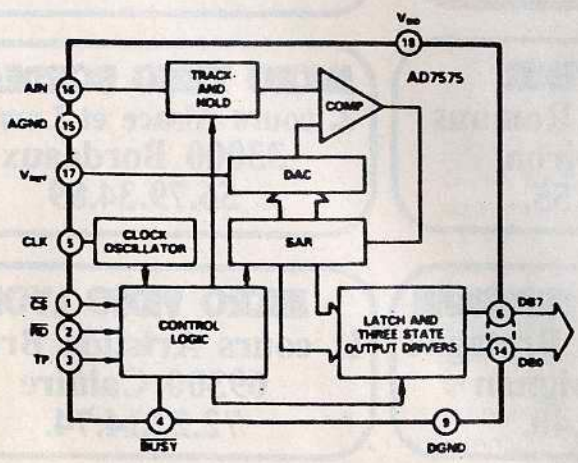


Fig.1: Diagrammes fonctionnels du convertisseur A/D

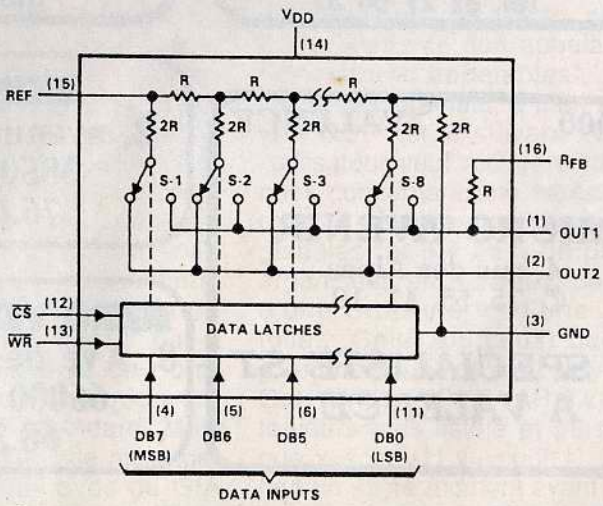


Fig.2: ...et du convertisseur D/A (source Analog Devices)

- La broche CS permet la validation du convertisseur lorsqu'elle passe à l'état bas ;
- La broche RD permet le lancement de la conversion lors de son passage à l'état bas ;
- La broche BUSY est à l'état bas tant que la conversion est en cours ;
- L'entrée CK permet de définir la fréquence de travail du convertisseur.

Grâce à son échantillonneur-bloqueur et à un temps de conversion très court (5 us), le circuit AD 7575 permet de digitaliser un signal analogique d'une amplitude de 2.46 volts crête à crête possédant une fréquence de 50 KHz. Ceci pour un prix de quelques dizaines de francs...

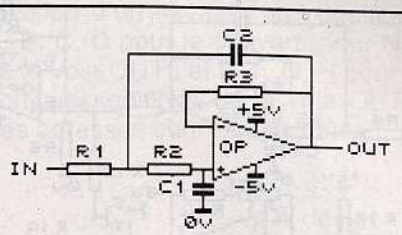
La figure 2 vous présente le diagramme fonctionnel du convertisseur N/A 8 bits, AD 7524 de Analog Devices, qui permet de convertir une valeur binaire variant entre $00 et $FF en un signal analogique proportionnel à cette valeur. Ce convertisseur utilise la conversion multiplicative. Ainsi, les bits DB0 à DB7 sont convertis en courant constant, à l'aide d'un réseau de résistances R-2R. Ces courants sont dirigés vers les sorties OUT1, OUT2 à l'aide d'interrupteurs électroniques dépendant de l'état des bits DB0-DB7. Les courants présents sur les sorties OUT1, OUT2 sont convertis en tension à l'aide d'un convertisseur courant/tension, réalisé avec un amplificateur opérationnel. La relation existant entre la valeur binaire (Data) et la tension en sortie (Vout) est :

Vout=-Vref*(Data/256),

où Vref est la tension de référence alimentant le réseau de résistance R-2R du convertisseur.

Ce circuit comporte plusieurs entrées de commandes qui sont :

- La broche CS permet la validation du convertisseur lorsqu'elle passe à l'état bas ;
- La broche WR permet le transfert de la valeur binaire présente sur les bits DB0-DB7 dans le buffer interne du convertisseur, lors de son passage à bas.



Si R1=R2=2xR3=R, la fréquence de coupure est égale à :

$$Fc = \frac{1}{R*\sqrt{C1*C2}}$$

avec C1, C2 en farads, R en ohms, et Fc en hertz

Fig.3: Structure du filtre passe-bas 40 dB/décade

Grâce à ces convertisseurs, il sera possible de concevoir une carte de conversion A/N et N/A permettant la digitalisation d'un signal analogique d'amplitude maximale de 2.5 volts avec une fréquence variant de 0 à 20 KHz.

PRINCIPE

La carte de conversion se compose de 2 modules distincts, avec un module de conversion A/N double et un module de conversion N/A double.

1) Le module de conversion A/N double

Il vous est présenté par la figure 4. Chaque voie d'entrée IN0, IN1 est composée d'un double filtre passe bas permettant de stopper tout signal dont la fréquence est égale ou supérieure à la moitié de la fréquence d'échantillonnage. Ceci pour éviter toutes distorsions lors de l'échantillonnage du signal. Ces filtres sont réalisés respectivement pour les voies IN0 et IN1 à l'aide des amplificateurs opérationnels OP1/OP2 et OP5/OP6. La structure du filtre ainsi que la formule de calcul de la fréquence de coupure est fournie par la figure 3. Cela vous permettra de recalculer, si nécessaire, cette fréquence en fonction de vos besoins.

Après le filtrage du signal, le sommateur réalisé respectivement pour les voies IN0 et IN1 autour des amplificateurs opérationnels OP3 et OP7 permet de travailler en mode unipolaire (cavaliers en position S) ou en mode bipolaire (cavaliers en

ULTIMA
INFORMATIQUE
5 Bd Voltaire 75011 PARIS Tél. (1) 43 38 96 31
72-74 rue de Paris 59800 LILLE Tél. 20 42 09 09
35 rue du Taur 31000 TOULOUSE Tél. 62 27 04 38

ATARI 520 STF et 1040 STF

PROMO D'ÉTÉ

520 STF
+ carte d'extension de 512Ko
2390F

1040 STF
2990F

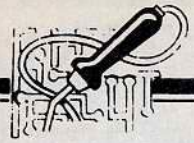
Pour les configurations avec moniteurs nous consulter.
Le matériel est neuf et livré avec manuel, souris, langage.

Matériel d'expo agence de Paris

▶ Moniteur SM124	890F
▶ Moniteur SC1224	1500F
▶ Méga ST1	4500F
▶ Portfolio	1500F
▶ ABC 286/30	6000F
▶ ABC 286/60H	8000F
▶ ABC 286/44R	10000F

Matériel d'expo STF et STE nous consulter

BON DE COMMANDE
à envoyer à ULTIMA/SARO VPC 5 Boulevard VOLTAIRE 75011 PARIS
Nom:
Prénom:
Adresse:
Code Postal:
Ville:
Tél:
Carte bleue n°:
Date d'expiration:
Signature:
Article(s) commandé(s):
Prix:
Port: +140F
Prix total:
Offre limitée Articles dans la limite des stocks disponibles ST07



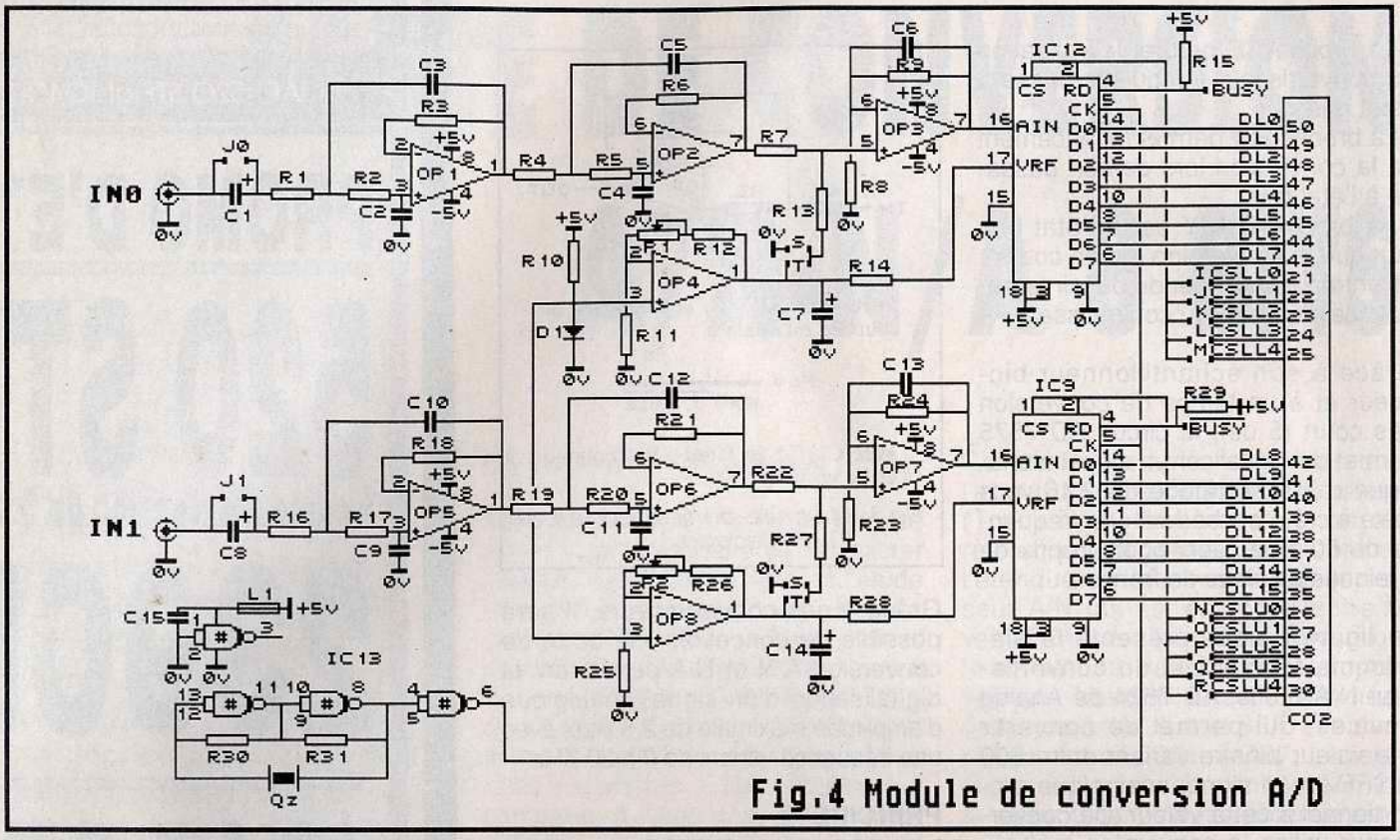


Fig.4 Module de conversion A/D

position T). Après ces traitements, le signal de chaque voie IN0-IN1 est fourni aux convertisseurs A/N respectifs (IC12-IC13). Chaque convertisseur a besoin d'une tension de référence qui lui est fournie par le montage réalisé autour des amplificateurs opérationnels OP4 pour la voie IN0 et OP8 pour la voie IN1. Cette tension de référence est ajustable, à l'aide des potentiomètres P1 pour la voie IN0 et P2 pour la voie IN1, entre 1 et 2.5 volts. Cela permettra à chaque voie de posséder une gamme d'entrée indépendante variant entre 0 volt et deux fois la tension de référence réglée.

Enfin les cavaliers J0 et J1 permettent de réaliser un commutateur AC/DC (courant alternatif/ courant continu). On enlèvera le cavalier lorsque l'on travaillera avec des signaux alternatifs afin d'enlever toute composante continue aux signaux à traiter.

La partie numérique est quant à elle très simple. Chaque convertisseur A/N se partage le bus de données de la carte interface DL0 à DL7 pour

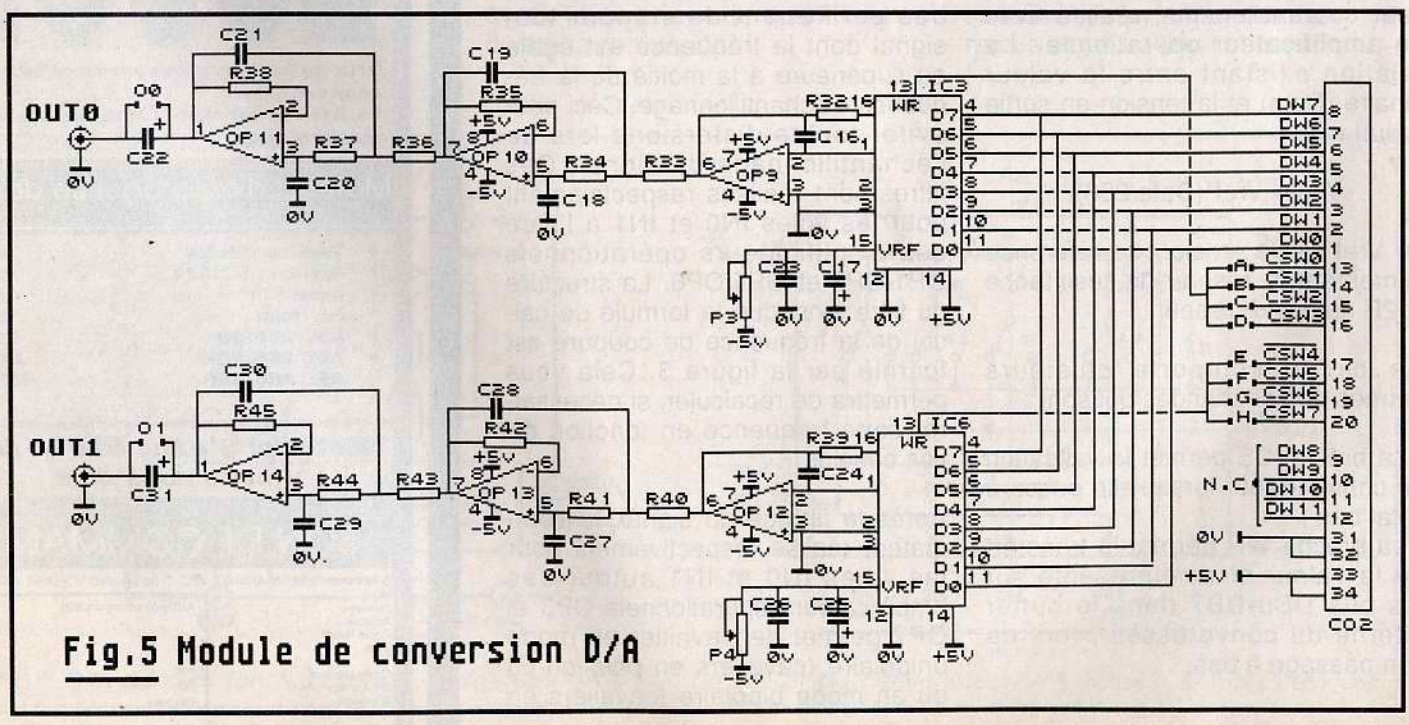


Fig.5 Module de conversion D/A

le convertisseur de la voie IN0, et DL8 à DL15 pour le convertisseur de la voie IN1. Cela permet la synchronisation de l'échantillonnage des signaux analogiques sur les deux voies. La lecture des données et le départ d'une nouvelle conversion s'effectuent lors du passage au niveau bas de l'un des signaux de validation CSLLn ou CSLUn. L'adresse de chaque convertisseur est définie par l'emplacement d'un cavalier aux positions I, J, K, L, M pour le convertisseur de la voie IN0 et N, O, P, Q, R pour le convertisseur de la voie IN1. Les adresses sont les suivantes :

Position du cavalier	Adresse validée
I	$FA 0001
J	$FA 0003
K	$FA 0005
L	$FA 0007
M	$FA 0009
N	$FA 0000
O	$FA 0002
P	$FA 0004
Q	$FA 0006
R	$FA 0008

Enfin une horloge, d'une fréquence de 4 MHz, réalisée autour de IC13, permet de cadencer le fonctionnement interne des deux convertisseurs et d'obtenir des temps de conversion de 5 us. Dans ce montage, les sorties BUSY des deux convertisseurs ne sont pas utilisées. Elles permettront de vérifier leur bon fonctionnement.

2) Le module de conversion N/A double

Le module de conversion N/A vous est présenté par la figure 5. Celui-ci se compose, comme pour le module A/N, de deux parties identiques. Chacun des convertisseurs N/A (IC3/IC6) est connecté en parallèle sur le bus de données de la carte interface DW0-DW7. La mémorisation dans le buffer de chaque convertisseur s'effectue lors du passage au niveau bas de l'un des signaux de validation CSWn.

Comme pour le module de conversion A/N, l'adresse de chaque convertisseur N/A est définie par la position d'un cavalier aux positions A, B, C, D pour le convertisseur N/A de la voie OUT0 et E, F, G, H pour le convertisseur N/A de la voie OUT1. Les adresses sont les suivantes :

Position du cavalier	Adresse validée
A	$FB 0000
B	$FB 0200
C	$FB 0400
D	$FB 0600
E	$FB 0800
F	$FB 0A00
G	$FB 0C00
H	$FB 0E00

Après la conversion de la valeur binaire en courant par les convertisseurs N/A, les amplificateurs opérationnels OP9, OP12 transforment ce courant en une tension proportionnelle. Ce signal est filtré au travers de deux filtres passe bas, réalisés autour des amplificateurs opérationnels OP10, OP11 pour la voie OUT0 et OP13, OP14 pour la voie OUT1. Ils permettent de lisser le signal et d'atténuer les fréquences supérieures ou égales à la moitié de la fréquence d'échantillonnage. La structure de ces filtres est identique à celle des filtres utilisés dans le module de conversion A/N.

Comme pour le convertisseur A/N, une tension de réference est nécessaire, celle-ci est ajustable grâce au potentiomètre P3 pour la voie OUT0 et P4 pour la voie OUT1 entre -1 volt et -4 volts. Cela permet d'obtenir une amplitude du signal de sortie indépendante et comprise entre 0 et -Vref sur les voies OUT0 et OUT1. Grâce aux cavaliers O0 et O1, il sera possible de fournir sur les voies OUT0, OUT1 un signal avec (cavalier présent) ou sans (cavalier absent) composante continue.

La figure 6 vous présente le brochage des différents circuits intégrés utilisés dans cette réalisation. La nécessité d'une tension négative (-5v), m'a amené à réaliser une alimentation symétrique +5v 0v -5v dont le schéma de principe vous est présenté par la figure 7. Elle remplacera avantageusement l'alimentation décrite lors du premier article.

AXE-3D
Promotions en juillet et en août
67, rue de la jonquière
75017 Paris - M° Guy Moquet
Tél: 42.28.08.39 ou 42.28.06.23
Ouvert du Lundi au Samedi
Bientôt, ouverture d'un magasin à La Roche sur Yon

ATARI STE

520 STE: **3490 F TTC**
1040 STE: **4490 F TTC**
STE étendu à 1040: **3680 F TTC**
STE étendu à 2060: **4790 F TTC**
STE étendu à 4160: **6200 F TTC**

En cadeau:
+ 70 Disquettes 3"1/2 DD DF!!!
+ 2 ans de garantie !!!

Extensions mémoires

SIMM ou SIPde 1 Mo: **699 F TTC**
SIMM ou SIP 256 ko: **260 F TTC**
Adaptateur SIMM/SIP: **15 F TTC**
Extension 512 ko pour 520 STF: installée: **499 F TTC**
Extension ST2-ST4 (2 Mo): installée: **1490 FTTC**

SUPER ! (pour STF, Mega ST1 et ST2)
Carte d'extension XSRAM 0/2: en 0 ko: **650 F TTC**
Carte d'extension XSRAM 0/4: en 0 ko: **780 F TTC**
Pack de 2 Mo pour XRAM: **1450 F TTC**

DIVERS

Moniteur SM 124: **1190 F TTC**
Moniteur Philips Couleur Stereo: **2290 F TTC**
Drive 3"1/2 720 ko ext.: **690 F TTC**
Synchro Express: **375 F TTC**
Boite de FreeWare I,II,III: **100 F TTC**
Disquettes 3"1/2 DD DF: **4,40 F TTC**

Star LC 10 serie II: **1790 F TTC**
Star LC 10 Couleur: **2100 F TTC**
Star LC 24/10: **2690 F TTC**
En cadeau: le cable!

Commandes VPC: port en sus.

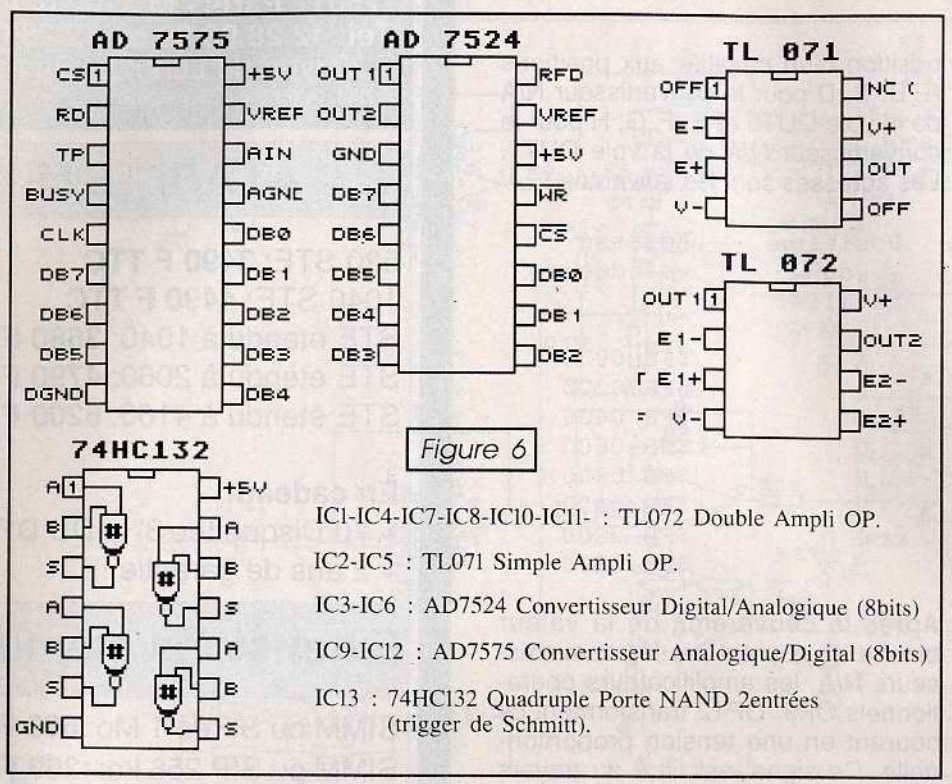


Figure 6

IC1-IC4-IC7-IC8-IC10-IC11- : TL072 Double Ampli OP.

IC2-IC5 : TL071 Simple Ampli OP.

IC3-IC6 : AD7524 Convertisseur Digital/Analogique (8bits)

IC9-IC12 : AD7575 Convertisseur Analogique/Digital (8bits)

IC13 : 74HC132 Quadruple Porte NAND 2entrées (trigger de Schmitt).

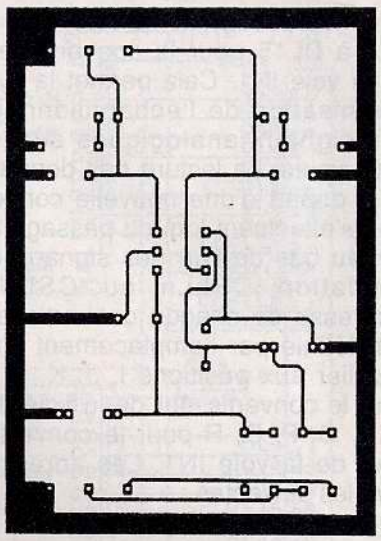


Fig.8 : Circuit imprimé de l'alimentation +-5V (Vue de dessus)

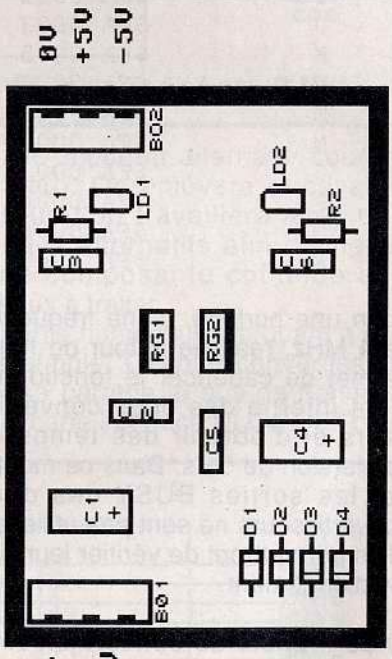


Fig.8 bis: Implantation des composants de l'alimentation +-5V

RÉALISATION PRATIQUE

Comme pour les cartes précédentes, le circuit imprimé a été réalisé en double face sur un support en époxy de format 155 x 100 mm. L'alimentation est réalisée, quant à elle, en simple face sur un support de dimensions 75x50 mm. Les figures 8, 9 et 10, vous présentent les typons à l'échelle 1, ils vous permettront la réalisation des circuits imprimés. La figure 11 vous présente l'implantation des composants sur la carte principale. Voici maintenant quelques précisions concernant le montage :

- Commencer par la réalisation du strap indiqué par le repère X sur la figure 11.
- Comme pour les cartes précédentes, il est indispensable d'utiliser des supports tulipes pour tous les circuits intégrés.
- Monter les supports des 13 circuits IC1 à IC13, en effectuant les soudures sur les deux faces du circuit imprimé.
- Monter les composants discrets (Résistances, Quartz, Selfs, Condensateurs, Bornier (BO) et les barrettes bas profil (BBP1, BBP2, BBP3, BBP4, BBP5, BBP6, J0, J1, O0, O1).
- Attention pour les composants L2, D1, C2, C4, C7, C9, C11, C17, C21, C30, C37, C38, C39, C40, R10, il faut effectuer les soudures sur les deux faces.
- Sur les entrées IN0, IN1 et les sorties OUT0, OUT1, on câblera, à l'aide de fils blindés, quatre fiches femelles de type RCA.
- Terminer par le montage du connecteur CO2.

Vérifier surtout qu'aucun court-circuit, dû à une mauvaise gravure ou à une mauvaise soudure, ne subsiste entre les pistes de la carte. Après ces

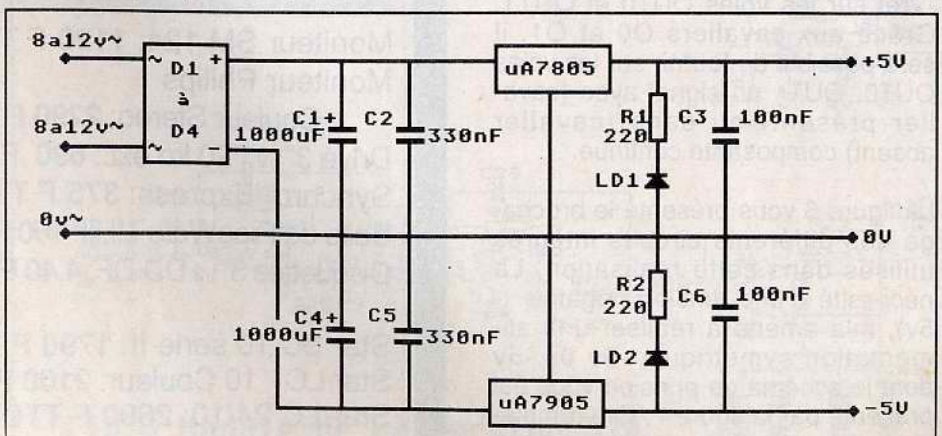


Fig 7: Schéma de principe de l'alimentation (+-5 V)

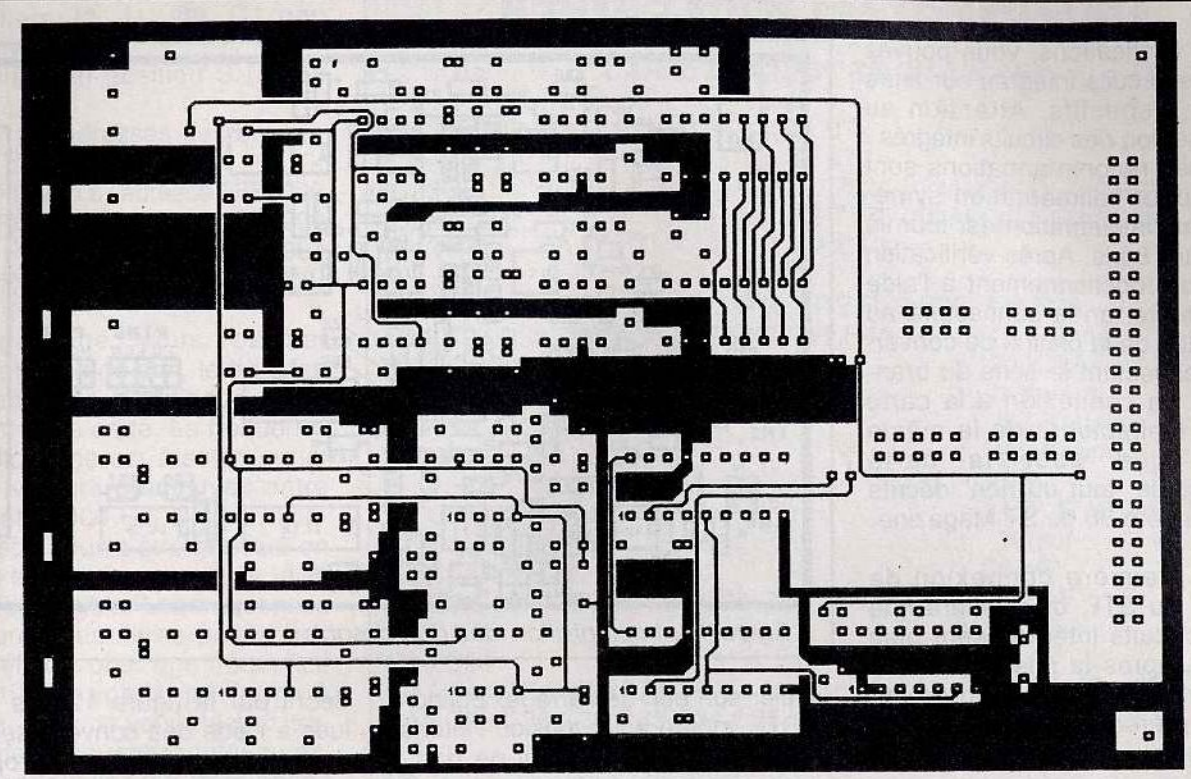


Figure 9 : Côté composants (Vue de dessus)

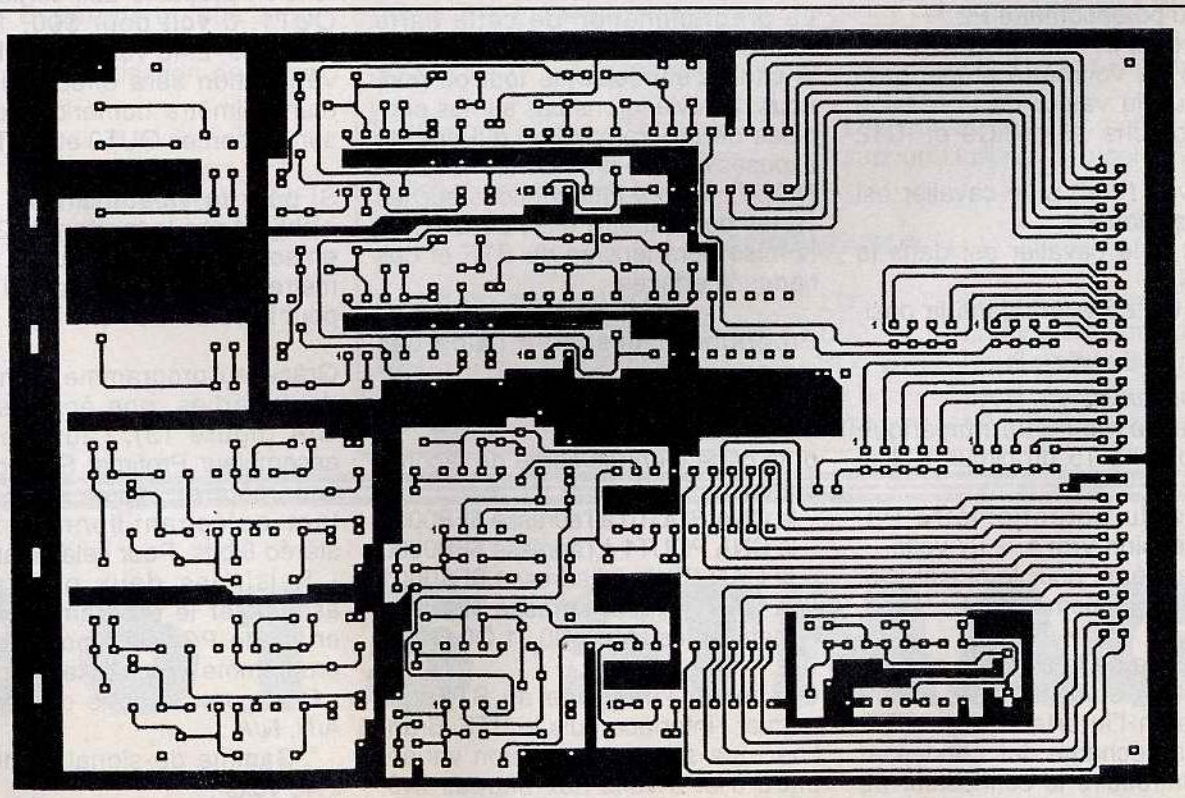


Figure 10 : Côté cuivre (toujours Vue de dessus)

quelques vérifications, vous pouvez insérer les circuits intégrés sur leurs supports respectifs. Attention au sens d'insertion des circuits intégrés. Les mêmes recommandations sont à suivre pour l'alimentation symétrique dont l'implantation est fournie par la figure 8 bis. Après vérification de son bon fonctionnement à l'aide d'un voltmètre, on la connectera au bornier (BO) de la platine de conversion en respectant le sens du branchement. La connexion à la carte interface s'effectuera de la même manière que pour la carte d'entrée/sortie "tout ou rien" décrite dans le numéro 38 de ST Magazine.

Avant la première connexion de l'interface au STF, on vérifiera que tous les circuits intégrés sont bien alimentés, après la mise sous tension des cartes. De plus on procédera aux préréglages des tensions de référence :

a) Des convertisseurs A/N
- Brancher un volmètre numérique entre la broche 17 du circuit IC 12 et la masse.
- A l'aide du potentiomètre P1, régler la tension Vref à 1.23 volt.
- Faire de même pour le circuit IC9, à l'aide du potentiomètre P2.
- Relier IN0 et IN1 à la masse.
- A l'aide du voltmètre numérique, vérifier que la valeur de la tension entre la broche 16 de IC9 et IC12 est de :
  * 1.23 volt (Vref) si le cavalier est dans la position T ;
  * 0 volt si le cavalier est dans la position S ;
- Laisser les cavaliers dans la position S.

b) Des convertisseurs N/A
- Brancher le voltmètre numérique entre la broche 15 du circuit IC3 et la masse.
- A l'aide du potentiomètre P3, régler la tension Vref à -2.27 volts.
- Faire de même pour le circuit IC6, à l'aide du potentiomètre P4.

Le préréglage de la carte de conversion A/N-N/A étant terminé, mettre hors tension l'interface et le STF, attendre la décharge des condensateurs, et introduire le connecteur de la carte interface dans le connecteur du port cartouche, allumer le STF, et

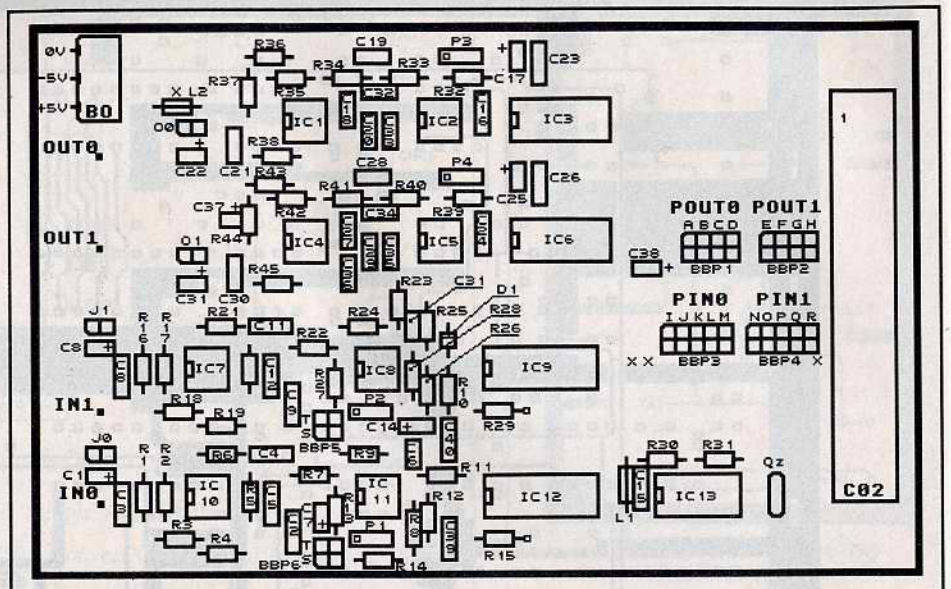


Figure 11 : Implantation des composants

vérifier son bon démarrage. Eteindre le STF, mettre sous tension l'interface et rallumer le STF, il ne doit y avoir aucune modification lors de son démarrage, et si c'est le cas nous pourrons passer au test logiciel de la carte.

TEST DE LA CARTE DE CONVERSION A/N-N/A

La programmation de cette carte s'effectuant de la même manière que la carte d'entrée/sortie tout ou rien, nous ne reviendrons pas sur les principes de programmation qui ont été exposés dans le n•38 de ST MAG.
NOTA : Toutes interventions sur les cartes devront être effectuées après la mise hors tension du STF et des cartes interfaces.

Programmer ensuite à l'aide des straps :
a) la carte interface E/S dans le mode 8 bits.
b) la carte d'entrée/sortie de façon à obtenir :
- le CNA POUT0 à l'adresse $FB0000
- le CNA POUT1 à l'adresse $FB0800
- le CAN PIN0 à l'adresse $FA0001
- le CAN PIN1 à l'adresse $FA0000
- Installer les straps J0,J1,O0,O1

Connecter l'interface au STF, alimenter l'interface puis le STF. Brancher une source de tension variant entre 0 et 3 volts aux entrées IN0, IN1 et régler celle-ci à 0 volt. Après avoir saisi et lancé le programme décrit par la figure 12, les valeurs lues à l'aide des convertisseurs A/N doivent varier de façon proportionnelle à la tension présente aux entrées IN0, IN1: $00 pour 0 volt, $80 pour 1.23 volt, $FF pour 2.46 volts. A chaque appui de la touche 'SPACE' la valeur affectée aux convertisseurs N/A est incrémentée, faisant varier proportionnellement la tension présente aux sorties OUT0, OUT1: 0 volt pour $00, 1.23 volt pour $80, 2.46 volts pour $FF. Cette vérification sera effectuée à l'aide d'un volmètre numérique connecté sur les sorties OUT0 et OUT1.

Si pour la valeur binaire $FF, on n'obtient pas la tension de 2.46 volts en sortie, il faut agir sur le potentiomètre P3 pour la sortie OUT0 et P4 pour la sortie OUT1.

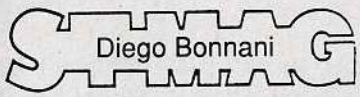
Grâce au programme composé de deux parties, une écrite en Basic GfA (figure 13), l'autre écrite en assembleur Profimat ST (figure 14), vous pourrez utiliser votre STF en tant qu'échantillonneur de son stéréo 8 bits. Pour cela, il faudra :
- Saisir les deux programmes, assembler le programme machine en mode PC-relatif pour obtenir un programme exécutable relogeable.
- Configurer la carte de conversion A/N, N/A :
  * Gamme de signal d'entrée 0 à 2.46 volts ;
  * Gamme de signal de sortie 0 à 2.46 volts ;

* Cavaliers J0, J1, O0, O1 non enfichés (Mode AC) ;
* Cavaliers en position S (Mode bipolaire) ;
* Définir les adresses des convertisseurs :
  le CNA POUT0 à l'adresse $FB0000
  le CNA POUT1 à l'adresse $FB0800
  le CAN PIN0 à l'adresse $FA0001
  le CAN PIN1 à l'adresse $FA0000.

Ce programme vous permet l'échantillonnage de son, le transfert des échantillons sur disque, ainsi que le test de la carte. La fréquence d'échantillonnage est à sélectionner parmi 12 valeurs prédéfinies entre 5000 Hz et 50000 Hz. L'échantillonnage du son pourra être effectué en mono ou en stéréo suivant la taille mémoire disponible. Les fréquences de coupures des filtres passe bas ont été définies pour une fréquence d'échantillonnage supérieure à 12000 Hz. Donc si vous désirez travailler à des fréquences inférieures, il faudra recalculer les fréquences de coupure des filtres.

CONCLUSION

Nous voici arrivés au terme de la série d'articles consacrée aux cartes d'entrées/sorties pour les STF 520/1040. Espérons qu'elle aura permis de vous ouvrir de nouvelles frontières dans l'utilisation de votre ordinateur favori...



NOMENCLATURE DES COMPOSANTS

**** CARTE DE CONVERSION A/N - N/A***

IC1,IC4,IC7,IC8,IC10,IC11	:	TL 072
IC2,IC5	:	TL 071
IC3,IC6	:	AD 7524 (ANALOG DEVICES)
IC9,IC12	:	AD 7575 (ANALOG DEVICES)
IC13	:	74HC132
R1,R2,R4,R5,R7,R8,R9,R13,R16, R17,R19,R20,R22,R23, R24,R27,R33,R34,R36,R37,R40,R41,R44	:	10 Kohms
R3,R6,R18,R21,R35,R38,R42, R45	:	20 Kohms
R12,R26,R32,R39	:	1 Kohms
R11,R15,R25,R29	:	4.7 Kohms
R14,R28	:	2.2 Kohms
C1,C8,C22,C31	:	4.7 uF 16 volts Tantale
C4,C11,C20,C29	:	180 pF
C2,C9,C18,C27	:	1 nF
C3,C5,C10,C12,C19,C21,C28,C30	:	4.7 nF
C6,C13	:	22 pF
C16,C24	:	12 pF
C7,C14,C17,C25,C37,C38	:	10 uF 16 volts
C15,C23,C26,C32,C33,C34,C35, C36,C39,C40	:	100 nF
D1	:	1N 4148
L1,L2	:	Selft 100 uH
P1,P2,P3,P4	:	Potentiomètres multitours 20 Kohms verticaux
Qz	:	Quartz 4 MHz
CO2	:	Connecteur 50 broches Mâle à 90•

BBP1-BBP2 : Barrettes bas profil à 2 rangées 4 contacts avec 2 straps femelles.
BBP3-BBP4 : Barrettes bas profil à 2 rangées 5 contacts avec 2 straps femelles.
BBP5-BBP6 : Barrettes bas profil à 2 rangées 2 contacts avec 1 strap femelle.

J0-J1-O0-O1 : Barrettes bas profil à 2 rangées 1 contact avec 1 strap femelle.

2 supports tulipes 18 broches
2 supports tulipes 16 broches
4 fiches femelles de type RCA
1 supports tulipes 14 broches
8 supports tulipes 8 broches

**** ALIMENTATION***

D1-D2-D3-D4	: 1N 4007	LD1	: Led rouge diam. 3 mm
C1-C4	: 1000 uF 25 volts	LD2	: Led verte diam. 3 mm
C2-C5	: 330 nF	BO1-BO2	: Borniers bas 3 bornes
C3-C6	: 100 nF	Rg1	: uA 7805
R1-R2	: 220 ohms	Rg2	: uA 7905

3615 STMAG

"FIGURE" 12 : LISTING GFA

```
' Initialisation des variables utilisées dans le programme
Cnar%=&HFB0000
Cnal%=&HFB0800
Canr%=&HFA0000
Canl%=&HFA0001
Fin%=0
Vl1%=0
'
Me1$="- VALEUR DE L'OCTET AFFECTE A LA CNA D'ADRESSE :"
Me2$="- VALEUR DE L'OCTET LU A LA CAN D'ADRESSE      :"
Me3$="- VALEUR DU MOT LU A lA CAN D'ADRESSE          :"
Bl$="      "
'
Cls
Print At(4,2),"* TEST DE LA CARTE DE CONVERSION A/N-N/A *"
Print At(8,22),"* TOUCHE 'SPACE' => INCREMENTATION OCTET"
Print At(8,23),"* TOUCHE 'F'     => ARRET DU TEST"
'
@Affiche_valeur_out
Void Peek(Cnar%+Vl1%*2)
Void Peek(Cnal%+Vl1%*2)
'
While Fin%<>1
  Touche$=Inkey$
  If Touche$=" " Then
    If Vl1%=256 Then
      Vl1%=0
    Endif
    @Affiche_valeur_out
    Void Peek(Cnar%+Vl1%*2)
    Void Peek(Cnal%+Vl1%*2)
    Inc Vl1%
  Endif
  Vi1%=Peek(Canr%)
  Vi2%=Peek(Canl%)
  Vi3%=Dpeek(Canr%)
  @Affiche_valeur_in
  If Touche$="F" Or Touche$="f" Then
    Vl1%=0
    Fin%=1
  Endif
Wend
Void Peek(Cnar%)
Void Peek(Cnal%)
'
Procedure Affiche_valeur_out
  Print At(67,10);Bl$
  Print At(67,11);Bl$
  Print At(5,10);Me1$;Hex$(Cnar%),Hex$(Vl1%)
  Print At(5,11);Me1$;Hex$(Cnal%),Hex$(Vl1%)
Return
'
Procedure Affiche_valeur_in
  Print At(67,14);Bl$
  Print At(67,15);Bl$
  Print At(67,16);Bl$
  Print At(5,14);Me2$;Hex$(Canr%),Hex$(Vi1%)
  Print At(5,15);Me2$;Hex$(Canl%),Hex$(Vi2%)
  Print At(5,16);Me3$;Hex$(Canr%),Hex$(Vi3%)
Return
```

"FIGURE" 13 : LISTING GFA

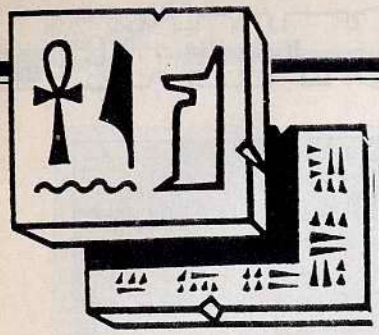
```
' ********************************************
' * Réservation de la mémoire pour le stockage *
' *    du code machine et des échantillons     *
' ********************************************
Dim A%(500),Pdiv%(20),Sdiv%(20),Pd%(8)
Taille%=Fre(0)-50000
Dim B%(Taille%/4)
Code%=Varptr(A%(0))
Len%=100000
Echan%=Varptr(B%(0))
F_echan%=Echan%+Len%
' ******************************
' * Chargement du code machine *
' ******************************
Bload "CODEMAC1.PRG",Code%
Debut%=Code%+28
Nfile$=""
Fin%=1
Md%=0
Infrq%=6
' ****************************************************
' * Initialisation des prédiviseurs et diviseurs     *
' * pour le MFP. Valeurs du prédiviseur PDIV% :1=>%4,*
' * 2=>%10,3=>%16,4=>%50,5=>%64,6=>%100,7=>%200      *
' * Valeur diviseur 0<DIV%<255                       *
' ****************************************************
Fqmfp=2457600
Pd%(1)=4
Pd%(2)=10
Pd%(3)=16
Pd%(4)=50
Pd%(5)=64
Pd%(6)=100
Pd%(7)=200
'
Pdiv%(0)=1
Pdiv%(1)=2
Pdiv%(2)=1
Pdiv%(3)=1
Pdiv%(4)=2
Pdiv%(5)=2
Pdiv%(6)=1
Pdiv%(7)=1
Pdiv%(8)=1
Pdiv%(9)=4
Pdiv%(10)=5
Pdiv%(11)=5
'
Sdiv%(0)=12
Sdiv%(1)=5
Sdiv%(2)=14
Sdiv%(3)=15
Sdiv%(4)=7
Sdiv%(5)=8
Sdiv%(6)=24
Sdiv%(7)=31
Sdiv%(8)=41
Sdiv%(9)=5
Sdiv%(10)=5
Sdiv%(11)=7
```

```
'
@Menu
@Set_frq
Hidem
While Fin%=1
  Key$=Inkey$
  Key%=Val(Key$)
  If Key%>0 And Key%<8 Or Key%=9 Then
    On Key% Gosub Sample,Replay,Monit,Save
    On Key%-4 Gosub Load,Mode,Set_frq,Rien,Fin
    Print At(15,14);String$(50," ");
    Print At(15,15);String$(50," ");
  Endif
Wend
Showm
End
Procedure Menu
  Cls
  Print At(17,2);"TEST DE LA CARTE DE CONVERSION"
  Print At(4,4);"--- FONCTIONS DISPONIBLES ---"
  Print At(4,6);"- ENREGISTREMENT D'UN ECHANTILLON..1"
  Print At(4,7);"- LECTURE D'UN ECHANTILLON.........2"
  Print At(4,8);"- TEST............................3"
  Print At(4,10);"- SAUVEGARDE D'UN ECHANTILLON.....4"
  Print At(4,11);"- CHARGEMENT D'UN ECHANTILLON.....5"
  Print At(45,6);"- MODE MONO/STEREO.....6"
  Print At(45,7);"- FREQUENCE D'ECHANT...7"
  Print At(45,8);"- FIN DE SESSION.......9"
  Print At(45,9);"- FREQUENCE D'ECHANT."
  Print At(45,10);" (5 Khz <Fe< 50 Khz) :"
  Print At(45,11);"- MODE...............:"
  Print At(1,12);String$(80,"_");
  Print At(68,11);"STEREO"
  @Aff_frq
Return
'
Procedure Set_frq
  Inc Infrq%
  If Infrq%>11 Then
    Infrq%=0
  Endif
  Af_frq%=Fqmfp/(Pd%(Pdiv%(Infrq%))*Sdiv%(Infrq%))
  Frq%=256*Sdiv%(Infrq%)+Pdiv%(Infrq%)
  @Aff_frq
Return
'
Procedure Aff_frq
  Print At(68,10);"              "
  Print At(68,10);Af_frq%;" Hz"
Return
'
Procedure Sample
  Print At(15,14);"Echantillonnage en cours..."
  Print At(15,15);"Retour au menu principal => 'ESC'"
  Lpoke Debut%+4,Echan%
  Lpoke Debut%+8,F_echan%
  Lpoke Debut%+12,Frq%
  Lpoke Debut%+16,0
  Lpoke Debut%+20,Md%
  Call Debut%
Return
'
Procedure Replay
  Print At(15,14);"Lecture en cours..."
  Print At(15,15);"Retour au menu principal => 'ESC'"
  Lpoke Debut%+4,Echan%
  Lpoke Debut%+8,F_echan%
  Lpoke Debut%+12,Frq%
  Lpoke Debut%+16,1
  Lpoke Debut%+20,Md%
  Call Debut%
Return
'
Procedure Monit
  Print At(15,14);"Monitoring en cours..."
  Print At(15,15);"Retour au menu principal => 'ESC'"
  Lpoke Debut%+4,Echan%
  Lpoke Debut%+8,F_echan%
  Lpoke Debut%+12,Frq%
  Lpoke Debut%+16,2
  Lpoke Debut%+20,Md%
  Call Debut%
Return
'
Procedure Save
  Fileselect "\*.*","",File$
  If File$<>"" Then
    Nfile$=File$
    Print At(15,14);"Sauvegarde en cours..."
    Bsave Nfile$,Echan%,Len%
  Endif
Return
'
Procedure Load
  Fileselect "\*.*",Nfile$,File$
  If File$<>"" Then
    Nfile$=File$
    Print At(15,14);"Chargement en cours..."
    Bload Nfile$,Echan%
  Endif
Return
'
Procedure Mode
  If Md%=0 Then
    Print At(68,11);"MONO  "
    Md%=1
  Else
    Print At(68,11);"STEREO"
    Md%=0
  Endif
Return
'
Procedure Rien
Return
'
Procedure Fin
  Fin%=0
Return
```

"FIGURE 14" : le listing assembleur

```
;****** REGISTRES DU MFP ******
MFP       EQU       $FFFA00
AER       EQU       MFP+$03
IERA      EQU       MFP+$07
IERB      EQU       MFP+$09
IPRA      EQU       MFP+$0B
ISRA      EQU       MPF+$0F
IMRA      EQU       MFP+$13
IMRB      EQU       MFP+$15
VECTEUR   EQU       MFP+$17
TACR      EQU       MFP+$19
TADR      EQU       MFP+$1F
INTVEC    EQU       $134
;****** REGISTRES DE L'ACIA ******
ACCONT    EQU       $FFFC00
ACDATA    EQU       $FFFC02
;****** PORT CARTOUCHE ******
CANR      EQU       $FA0000   ; can voie IN0
CANL      EQU       $FA0001   ; can voie IN1
CNAR      EQU       $FB0000   ; can voie OUT0
CNAL      EQU       $FB0800   ; can voie OUT1
;****** FONCTIONS GEMDOS ******
F_SUPER   EQU       $20
;****** FONCTIONS XBIOS ******
ID_CLV    EQU       $19
;****** PROGRAMME ******
 TEXT
          bra       run
;****** ZONE DE DONNEES ******
dt:
s_deb:      ds.l  1   ; zone de transfert
s_fin:      ds.l  1   ; des valeurs fournies
s_freq:     ds.l  1   ; par le basic GFA
s_ftc:      ds.l  1   ; 5 longs mots
s_md:       ds.l  1   ;
reserve:    ds.l  8

spile:      ds.l  1          ; sauvegarde pile utilisateur
mfpmem:     ds.b  8          ; zone sauvegarde registres MFP
 ALIGN.W
deval_clv:  dc.b  $12,$1a      ;Chaîne inhibition souris/joy.
 ALIGN.W
vald_clv:   dc.b  $08,$00      ;Chaîne validation souris/joy.
 ALIGN.W
fin_int:    ds.b  1            ;flag fin interruption
 ALIGN.W
;****** DEBUT DU PROGRAMME ******
run:
  movem.l  a0-a6/d0-d7,-(sp)  ;sauvegarde contexte
  bsr      inib_clv           ;inhibition souris/joystick
  clr.l    -(sp)
  move.w   #F_SUPER,-(sp)
  trap     #$01
  addq.l   #$06,sp            ;passage en mode superviseur
  lea      dt(pc),a1
  move.l   d0,spile-dt(a1)    ;sauvegarde pile utilisateur
  bsr      sauve_mfp          ;sauvegarde registres du MFP
  move.w   #$2700,sr          ;inhibition des interruptions
  lea      int_nul(pc),a0
  bsr      setint             ;installation vecteur d'IT MFP
  move.l   s_freq-dt(a1),d0
  bsr      set_freq           ;installation fréquence d'éch.
  bsr      val_mfp            ;validation du MFP
  move.w   #$2500,sr          ;validation IT MFP

  move.l   s_deb-dt(a1),a6    ;a6=début échantillon
  move.l   s_fin-dt(a1),d6    ;d6=fin échantillon
  lea      CANR,a2            ;affectation des
  lea      CANL,a3            ;adresses
  lea      CNAR,a4            ;des convertisseurs
  lea      CNAL,a5            ;
  move.w   #$00ff,d5          ;d5=masque d'isolement

  st       fin_int-dt(a1)     ;flag fin interruption=$FF
  move.l   s_ftc-dt(a1),d0
  cmp.l    #$02,d0
  bgt      fin                ;si N• fonction >2 sortir
  bne      samp_rep
  sf       fin_int-dt(a1)     ;si N•fonction=2 alors
                              ;flag fin interruption=$00
samp_rep:
  tst.l    s_md-dt(a1)
  beq      md_stereo          ;si mode stéréo
  cmp.l    #$00,d0            ;si mode mono
  bne      cx1
  lea      m_sample(pc),a0    ;et N• fonction=0 alors
  bra      set_int            ;enregistrer
cx1:
  cmp.l    #$01,d0
  bne      cx2
  lea      m_replay(pc),a0    ;et N•fonction=1 alors lire
  bra      set_int
cx2:
  lea      m_monit(pc),a0     ;et N•fonction=2 alors test
  bra      set_int

md_stereo:
  cmp.l    #$00,d0
  bne      cx11
  lea      s_sample(pc),a0
  bra      set_int
cx11:
  cmp.l    #$01,d0
  bne      cx12
  lea      s_replay(pc),a0
  bra      set_int
cx12:
  lea      s_monit(pc),a0

set_int:
  bsr      setint        ;installation du vecteur d'IT MFP
                         ;en fonction de l'opération
boucle:
  tst.b    fin_int-dt(a1)  ;tâche de fond
  bne      boucle          ;attente de fin de l'IT
  btst     #$00,ACCONT
  beq      boucle
  move.b   ACDATA,d0
  cmp.b    #$01,d0
  bne      boucle          ;attente d'appui sur 'Esc'

fin:
```

```
  move.w   #$2700,sr        ;inhibition des IT
  bsr      rest_mfp         ;restitution registres du MFP
  bsr      rest_clv         ;validation souris/joystick
v_clv:
  btst     #$00,ACCONT
  beq      suite
  move.b   ACDATA,d0
  bra      v_clv            ;vide le tampon clavier
suite:
  move.w   #$2000,sr        ;validation de toutes les IT
  lea      dt(pc),a1
  move.l   spile-dt(a1),-(sp)
  move.w   #F_SUPER,-(sp)
  trap     #$01
  addq.l   #$06,sp            ;retour mode utilisateur
  movem.l  (sp)+,a0-a6/d0-d7 ;restitution contexte
  rts                         ;retour au basic
;****** Routines d'interruptions ******
int_nul:  rte
m_replay:
  move.b   (a6)+,d7           ;lecture échantillons mono
  cmp.l    d6,a6
  bgt      fin_intp
  and.w    d5,d7
  add.w    d7,d7
  move.b   0(a4,d7.w),d7
  rte
fin_intp:
  move.w   #$2700,sr          ;fin IT
  lea      int_nul,a0
  bsr      setint
  sf       fin_int-dt(a1)
  move.w   #$2500,sr
  rte
s_replay:
  move.b   (a6)+,d7           ;lecture échantillons stéréo
  move.b   (a6)+,d4
  cmp.l    d6,a6
  bgt      fin_intp
  and.w    d5,d7
  and.w    d5,d4
  add.w    d7,d7
  add.w    d4,d4
  move.b   0(a4,d7.w),d7
  move.b   0(a5,d4.w),d4
  rte
m_sample:
  move.b   (a2),(a6)+         ;enregistrement échant. mono
  cmp.l    d6,a6
  bgt      fin_intp
  rte
s_sample:
  move.w   (a2),(a6)+         ;enregistrement échant. stéréo
  cmp.l    d6,a6
  bgt      fin_intp
  rte
m_monit:
  move.b   (a2),d7       ;test mono
  and.w    d5,d7
  add.w    d7,d7
  move.b   0(a4,d7.w),d7
  rte
s_monit:
  move.b   (a2),d7       ;test stéréo
  move.b   (a3),d4
  and.w    d5,d7
  and.w    d5,d4
  add.w    d7,d7
  add.w    d4,d4
  move.b   0(a4,d7.w),d7
  move.b   0(a5,d4.w),d4
  rte
;************************
sauve_mfp:
  move.b   IERA,mfpmem-dt(a1)        ;sauvegarde registres
  move.b   IERB,mfpmem+1-dt(a1)      ;du MFP
  move.b   IMRA,mfpmem+2-dt(a1)
  move.b   IMRB,mfpmem+3-dt(a1)
  move.b   TADR,mfpmem+4-dt(a1)
  move.b   TACR,mfpmem+5-dt(a1)
  move.b   VECTEUR,mfpmem+6-dt(a1)
  rts
rest_mfp:
  move.b   mfpmem-dt(a1),IERA        ;restitution registres
  move.b   mfpmem+1-dt(a1),IERB      ;du MFP
  move.b   mfpmem+2-dt(a1),IMRA
  move.b   mfpmem+3-dt(a1),IMRB
  move.b   mfpmem+4-dt(a1),TADR
  move.b   mfpmem+5-dt(a1),TACR
  move.b   mfpmem+6-dt(a1),VECTEUR
  rts
setint:
  move.w   sr,d0              ;affectation de l'adresse de
  move.w   #$2700,sr          ;la routine d'interruption
  move.l   a0,INTVEC          ;au vecteur IT MFP
  move.w   d0,sr
  rts
inib_clv:
  pea      deval_clv(pc)      ;dévalidation souris/joy.
  move.w   #$01,-(sp)
  bra      ftc_clv
rest_clv:
  pea      vald_clv(pc)       ;validation souris/joy.
  move.w   #$00,-(sp)
ftc_clv:
  move.w   #ID_CLV,-(sp)
  trap     #$0e
  addq.l   #$08,sp
  rts
set_freq:
  move.b   #0,TACR            ;réglage de la fréquence
  and.l    #$0000ffff,d0      ;d'échantillonnage
  move.b   d0,TACR
  lsr.w    #8,d0
  move.b   d0,TADR
  rts
val_mfp:
  move.b   #$20,IMRA       ;mise en route du MFP
  move.b   #$00,IMRB
  move.b   #$20,IERA
  move.b   #$00,IERB
  bclr     #$03,VECTEUR
  rts
 END
```



LE SOUNDTRACKER ET LE ST

Le Soundtracker est l'un des programmes de musique les plus utilisés sur l'Amiga. Il permet de composer des musiques à l'aide d'instruments digitalisés, le son étant par conséquent rendu plus réaliste. Cet article et le source en assembleur l'accompagnant vous permettront de mieux comprendre le principe du Soundtracker et de son émulation sur ST.

Il faut savoir en effet que les modules n'ont aucunement besoin d'être modifiés pour être rejoués par le ST : il vous suffit de les transférer de l'Amiga sur le ST à l'aide d'un logiciel comme Dos-2-Dos (Central Coast Software).

LES ECHANTILLONS SONORES

Pour une définition de l'échantillon sonore, nous vous invitons à vous reporter au numéro 41 de ST Mag, page 96. Rappelons simplement que ceux-ci sont stockés en mémoire sous forme d'octets. Un son, quel qu'il soit, est toujours défini par trois paramètres :

Le timbre : il s'agit de la vibration de base dans un son. Il détermine sa nature et lui est propre. Par exemple le Commodore 64 propose cinq types de timbres : sinusoïdale, triangulaire, dents de scie, carrée et bruit blanc.

La fréquence : c'est le nombre de vibrations par seconde. La fréquence est exprimée en hertz (inverse du temps d'une vibration). Elle détermine, par conséquent la hauteur d'un son. Notons que la bande passante de réception de l'oreille humaine va de 30 Hz à 18000 Hz chez un homme jeune (en moyenne).

L'intensité : elle représente le volume du son. La zone audible s'étend de 1 à 120 dB (l'intensité normale d'un son est de 80 dB).

On vient de voir plus haut que la fréquence définit la hauteur d'un instrument. Par exemple, la note La est à une fréquence de 440 Hz. Dans une octave, on dénombre 12 demi-tons qui sont les suivants: Do (C), Do dièse (C#), Ré (D), Ré dièse (D#), Mi (E), Fa (F), Fa dièse (F#), Sol (G), Sol dièse (G#), La (A), La dièse (A#), Si (B). Pour passer d'un demi-ton au suivant (ou au précédent), on multiplie (ou on divise) la fréquence par $2^{1/12}$. Il faut bien évidemment répéter l'opération autant de fois qu'il y a d'étapes (exemple: en passant du La au Sol, on multipliera la fréquence 3 fois par $2^{1/12}$).

Voici les fréquences de quelques notes:

Note	Fréquence
C	261.7
C#	277.2
D	293.7
D#	311.2
E	329.7
F	349.3
F#	370.0
G	392.0
G#	415.3
A	440.0
A#	466.2
B	493.9
C	523.3

En général, les instruments du Soundtracker Amiga sont digitalisés à une fréquence de 8.4 Khz, pour un Do de fréquence 261.7 Hz (on ne reviendra pas sur la distinction qu'il y a à faire entre la notion de fréquence d'échantillonnage et celle de fréquence d'une note).

Sur l'Amiga, c'est Paula qui se charge de la restitution des sons. Pour lui faire jouer un sample à une fréquence donnée, il faut lui transmettre le nombre de cycles d'attente

entre la restitution de deux octets consécutifs: c'est la période. La durée d'un cycle de Paula est de $2{,}79365.10^{-7}$ secondes. La fréquence de restitution (fr) se calcule donc en fonction de la période (t) de la manière suivante:

$$fr = \frac{1}{t * 2{,}79365.10^{-7}}$$

Par exemple, pour une fréquence de restitution de 8.4 Khz, on trouve une période égale à 428 ($1AC). Nous vous invitons à vous reporter à la Bible de l'Amiga pour plus de détails (éd. Micro-Applications).

On en déduit la table de correspondance suivante, table qui est d'ailleurs propre au Soundtracker Amiga:

Note	Octave 1	Octave 2	Octave 3
C	$358	$1AC	$0D6
C#	$328	$194	$0CA
D	$2FA	$17D	$0BE
D#	$2D0	$168	$0B4
E	$2A6	$153	$0AA
F	$280	$140	$0A0
F#	$25C	$12E	$097
G	$23A	$11D	$08F
G#	$21A	$10D	$087
A	$1FC	$0FE	$07F
A#	$1E0	$0F0	$078
B	$1C5	$0E2	$071

LA RESTITUTION DE PLUSIEURS SAMPLES SIMULTANEMENT

Le Soundtracker de l'Amiga restitue des musiques sur 4 voies. Il faut savoir avant toute chose qu'il ne s'agit pas, en les jouant sur le ST, de rajouter une voie aux trois autres existant déjà: rappelons en effet que lorsque le ST joue un sample, ses trois voies sont occupées (cela est dû à la conception même du Yamaha).

Sur le ST, il va donc falloir faire faire le gros du travail par le 68000: celui-ci va en fait mâcher le travail pour le Yamaha, qui se charge bien sûr de la restitution sonore en elle-même.

N'importe quel son de la nature voyage dans l'air sous forme de vibrations. Lorsque 2 sons parviennent en même temps à l'oreille humaine, leurs vibrations sont additionnées pour n'en former qu'une; le cerveau humain se charge alors de les séparer. Dans le cas du ST, tout va se passer de la même façon: le 68000 va, à partir des samples, fabriquer l'onde finale en procédant à leur addition, c'est-à-dire que les 4 voies ne seront en fait restituées que sous la forme d'un sample. Il n'y a donc pas de miracle: les 4 voies ne sont pas restituées indépendamment les unes des autres.

Pour le ST-E, c'est différent: en effet, sur cette machine, il est possible de jouer deux samples simultanément (un à gauche et un à droite). Sur chaque voie, on ne procédera donc qu'à l'addition de deux samples seulement, au lieu de quatre sur les ST, sauf si on veut faire un Soundtracker huit voies (Oktalyzer)!

LES VARIATIONS DE VOLUME

Il s'agit d'un aspect non négligeable du Soundtracker, puisque 2 commandes interviennent sur le volume d'un sample (Commandes A et C, voir plus loin). Les meilleures musiques Soundtracker (qui sont toujours les plus récentes!) utilisent cette possibilité de variation de volume pour rendre des effets des plus réussis. A nous donc d'émuler ces variations de volume sur ST...

Il faut distinguer entre le volume général du sample et l'enveloppe. Celle-ci peut se définir comme représentant les niveaux instantanés de volume (intensité, voir plus haut). En changeant l'enveloppe, on peut obtenir des sons radicalement différents, à partir d'un même sample (cf. Mad Max).

Sur le Soundtracker, on peut difficilement intervenir sur l'enveloppe, mais la commande A (installée à partir de la version 2.3 par Mnemotron de Spreadpoint) en permet toutefois une approche.

Sur l'Amiga, il y a 65 niveaux de volume (de 0 à 64 inclus) par voie. Le décibel (dB) est l'unité de mesure couramment utilisée; en schématisant, il représente le rapport entre le niveau de sortie d'un son et son niveau de référence. On trouvera ci-après un tableau des correspondances en dB pour chaque niveau de volume de l'Amiga.

Niveau	dB	Niveau	dB	Niveau	dB	Niveau	dB
1	-36.1	17	-11.5	33	-5.8	49	-2.3
2	-30.1	18	-11.0	34	-5.5	50	-2.1
3	-26.6	19	-10.5	35	-5.2	51	-2.0
4	-24.1	20	-10.1	36	-5.0	52	-1.8
5	-22.1	21	-9.7	37	-4.8	53	-1.6
6	-20.6	22	-9.3	38	-4.5	54	-1.5
7	-19.2	23	-8.9	39	-4.3	55	-1.3
8	-18.1	24	-8.5	40	-4.1	56	-1.2
9	-17.0	25	-8.2	41	-3.9	57	-1.0
10	-16.1	26	-7.8	42	-3.7	58	-0.9
11	-15.3	27	-7.5	43	-3.5	59	-0.7
12	-14.5	28	-7.2	44	-3.3	60	-0.6
13	-13.8	29	-6.9	45	-3.1	61	-0.4
14	-13.2	30	-6.6	46	-2.9	62	-0.3
15	-12.6	31	-6.3	47	-2.7	63	-0.1
16	-12.0	32	-6.0	48	-2.5	64	0

Sur le ST, et à l'aide de ce tableau, nous avons dressé une table comportant les valeurs de chaque octet d'un sample pour chaque niveau de volume de l'Amiga.

LE FORMAT D'UN MODULE

Format d'un module Soundtracker à 15 instruments (les valeurs en hexadécimal sont precédées d'un "$"):
- En-tête (nom de la musique et informations instruments)
- Séquence (Ordre des patterns à jouer)
- Partition (les patterns)
- Sons (samples).

A noter que toutes les adresses données ci-après seront relatives au début du module. Par exemple: l'octet MOD+$200 représente le 512ème octet du module.

En-tête

• MOD + $0: 20 octets qui contiennent le nom de la musique en ASCII.

• MOD + $14: 30 octets autant de fois qu'il y a d'instruments, c'est-à-dire 15 fois. Voici le détail du contenu de ces 30 octets:

22 octets pour le nom de l'instrument.
2 octets pour la longueur (length), exprimée en mots.
2 octets pour le volume (de $00 à $40).
2 octets pour le repeat: une fois que le sample a été joué sur toute sa longueur, le Soundtracker le rejoue à partir de la valeur indiquée par ces deux octets. Cette valeur est aussi exprimée en mots.
2 octets pour le repeat length: longueur à rejouer lors du repeat. Si elle est à 1, le Soundtracker boucle sur le même octet: le sample s'arrête. Valeur exprimée en mots.

Séquence

• MOD + $1D6: 1 octet. Nombre de positions (nombre de patterns à utiliser pour jouer la musique une seule fois). Le maximum est fixé à 128 patterns ($7F).

• MOD + $1D7: valeur souvent égale à $78, mais ne correspond à rien dans la version actuelle.

• MOD + $1D8: 128 octets correspondant aux numéros des patterns à jouer. Ex: 00, 00, 01, 01, 02, 02, 02. Les patterns à jouer seront donc les suivants: 0, 0, 1, 1, 2, 2, 2.

Partition

• MOD + $258: au minimum un pattern de 1 Ko, au maximum 128 patterns. Un pattern a 64 lignes, chacune correspondant aux informations des voies 0, 1, 2 et 3. Un pattern est égal à 1024 octets en mémoire, donc une ligne vaut 16 octets, et une voie, 4 octets.
Ces 4 octets sont de la forme suivante:
2 octets pour la note, comprise entre $71 et $358: c'est la période Amiga.
2 octets pour l'instrument et la commande, organisés sous la forme suivante: $XYZZ.

X représente le numéro de l'instrument, entre $1 et $F. Si cette valeur est égale à 0, c'est l'instrument précédemment utilisé qui sera réutilisé.
Y représente le numéro de la commande: 0, 1, 2, 3, 4, A, B, C, D, E, ou F.
Enfin, ZZ représente l'octet de paramètre de la commande.

Nous n'allons détailler, dans les paragraphes suivants, que les commandes implémentées jusqu'à la version 2.5. On sait en effet que le Noisetracker 2.0, qui n'est pas encore commercialisé, en comportera de nouvelles. On dénombre donc les commandes suivantes:

$0 = Aucun effet, ou arpège
$1 = Portamento Up
$2 = Portamento Down
$3 = Tone-Portamento
$4 = Vibrato
$A = Volume Slide
$B = Position Jump
$C = Set Volume
$D = Pattern Break
$E = Set Filter
$F = Set Speed

Etudions-les plus en détail maintenant, après avoir noté (sous forme de rappel), qu'un mot vaut deux octets.

• 0: aucun effet, ou arpège. Si l'octet de paramètre est différent de zéro, alors un arpège sera joué. Premier quartet: premier incrément en fréquence. Deuxième quartet: deuxième incrément en fréquence. L'arpège consiste à faire varier la note jouée à l'aide de ces deux quartets.

• 1: Portamento Up. La valeur (de $00 à $FF) indique la vitesse de l'incrémentation de la fréquence.

• 2: Portamento Down. La valeur (de $00 à $FF) indique la vitesse de la décrémentation de la fréquence.

• 3: Tone-Portamento. La note précédant la commande correspond à la note d'arrivée et l'octet de commande, la vitesse à laquelle le passage se fait à partir de l'ancienne note.

• 4: Vibrato. Le premier quartet représente la vitesse du vibrato, le deuxième son amplitude. Si on veut continuer l'effet, il suffit de remettre la commande 4 seule.

• A: Volume Slide. L'effet est ici le même que pour le portamento, sauf qu'il s'agit du volume. La représentation est toutefois différente: Si le quartet supérieur est non-nul, sa valeur indique la vitesse à laquelle le volume est augmenté, et inversement, c'est-à-dire que si c'est le quartet inférieur qui est non-nul, sa valeur indiquera la vitesse à laquelle le volume est baissé. Dans le cas où les deux valeurs sont non-nulles, c'est le quartet supérieur qui sera pris en compte: Le volume sera donc augmenté.

• B: Position Jump. L'octet indique le numéro de la séquence qui sera jouée à partir de la prochaine lecture de note.

• C: Set Volume. L'octet indique le volume auquel doit être joué l'instrument.

• D: Pattern Break. Le pattern est coupé, et on passe à la séquence suivante.

• E: Set Filter. Il n'a pas été implémenté dans cette version; il s'agit d'une commande qui ne fonctionne que sur certains A500 et A2000, et de toutes façons pas sur les A1000. Son utilisation est par ailleurs

déconseillée par les auteurs du programme. $E01 = Filtre et Power' LED déconnectés. $E00 = Filtre et Power LED connectés.

• F: Set Speed. L'octet de paramètre donne la nouvelle vitesse: nombre de 50ème de secondes (VBL) s'écoulant entre deux lectures de notes consécutives. La valeur par défaut est $F06, mais elle peut aller de $F01 (rapide) à $F1F (lente). Jusqu'à la version 2.4, la vitesse la plus lente était de $0F.

Pour tenter de clarifier cette partie qui peut sembler ardue, nous allons prendre un exemple: C#3 1C25 signifie que l'instrument 1 va être joué à la note C#3, et à un volume de 25. En mémoire, on aura: $00CA 1C25.

Si vous voulez mieux comprendre comment toutes ces commandes fonctionnent, vous pouvez modifier quelques modules vous-mêmes, et ainsi vous rendre compte des effets que le Soundtracker peut reproduire.

Sons

On trouve l'adresse du premier sample par un calcul relativement simple: MOD + $258 + Nbre de patterns * 1024.

Nbre de patterns correspond au total des patterns utilisés par la musique. Comme on connaît la longueur du premier sample grâce à l'en-tête, il suffit de le rajouter à l'adresse précédemment calculée, et on obtient l'adresse du second sample, etc.

Nous en arrivons maintenant à la différence avec les modules à 31 instruments. On les reconnaît grâce à MOD + $438, où se trouve la chaîne de caractère M.K. (Mahoney et Kaktus, les deux suédois de Northstar, fameux auteurs de Noisetracker).

Différences:

• Dans l'en-tête, on trouve de la place pour 31 instruments (au lieu de 15).

• La séquence sera: MOD + $3B6 pour le nombre de positions et MOD + $3B8 pour la table des patterns à jouer.

• On trouve la partition à l'adresse MOD + $43C. Les patterns sont légèrement différents: dans la partition, les 2 octets de la note contiennent dans le bit 24, le bit 5 du numéro de l'instrument. S'il est positionné à 1, il faut rajouter 16 au numéro de l'instrument. Ex: $1|358, $1000 -> Instrument 17, note C, octave 1 (représentation en mémoire).

LE SOURCE ST

Le source qui accompagne cet article a été programmé avec Devpac 2 (HiSoft). Il est relogeable, laisse tous les registres de libres, et un petit programme l'utilisant est fourni en exemple. Les fichiers binaires sont calculés à l'aide de programmes écrits en GfA Basic.

Le premier de ces programmes permet de calculer une table de fréquence traduisant les périodes envoyées à Paula. A% est le facteur de prédivision du Timer A du MFP. B% est la valeur du Data Register du Timer A (FRQ_DATA dans le source Assembleur prinipal). Ici, les deux valeurs sont fixées pour une fréquence de restitution de 14 Khz.

Le second programme permet de recalculer les samples à différentes intensités.

La routine jointe à cet article privilégie la qualité sonore et la facilité d'intégration dans vos propres programmes Assembleur. Cette version n'est pas la plus rapide que nous ayons faite (loin de là), mais elle devrait suffire pour la plupart des applications.

Une dernière chose: vous pourrez bientôt composer des musiques Soundtracker directement sur votre ST, plusieurs éditeurs semblables à celui de l'Amiga étant en préparation, dont certains en Domaine Public.

Klaus BERG
Stéphane CATALA
Fabien MASSON

```
; Listing GfA 1
Hidem
A%=4
B%=44
Buffer$=Space$(856*4)
Frq=(1/(2.79365*10^-7))/(2.4576*10^6/(A%*B%))
For I=1 To 856
  Lpoke Varptr(Buffer$)+(I-1)*4,65536*Frq/I
Next I
Bsave "A:\14_KHZ.FRQ",Varptr(Buffer$),856*4
Showm
```

```
; Listing GfA 2
Buf$=Space$(64*256)
For B%=0 To 255
  Poke Varptr(Buf$)+B%,128
Next B%
For T%=63 Downto 1
  Read V%
  For B%=0 To 255
    Poke Varptr(Buf$)+T%*256+B%,(B%-128)/Exp(V%/100)+128
  Next B%
Next T%
Bsave "A:\VOLUME.BIN",Varptr(Buf$),64*256
Data 0,1,3,4,6,7,9,10,12,13,15,16,18,20,21,23
Data 25,27,29,31,33,35,37,39,41,43,45,48,50,52,55,58
Data 60,63,66,69,72,75,78,82,85,89,93,97,101,105,110,115
Data 120,126,132,138,145,153,161,170,181,192,206,221,241
Data 266,301
```

```
; Routine écrite par Fabien MASSON et
; Stéphane CATALA.
; Voici quelques renseignements addition-
; nels sur la routine et ses buffers, qui
; vous permettront de mieux comprendre le
; programme:
;
; VOICEx
; voice+$0 (1 long mot): dernière ligne lue
;         dans la partition
; voice+$4(1 long mot): adresse de l'ins-
;         trument utilisé
; voice+$8 (1 mot): longueur de l'instru-
;         ment en mots
; voice+$A (1 long mot): adresse de la
;         partie du Sample a répéter
; voice+$E (1 mot): repeat length en mots
; voice+$10 (1 mot): période (=note)
; voice+$12 (1 mot): volume
; voice+$14 (1 mot): inutilisé
; voice+$16 (1 octet): pour la commande 3
; voice + $17 (1 octet): pour la commande 3
; voice + $18 (1 mot): période (=note)
;         d'arrivée pour la commande 3
; voice + $1A (1 octet): pour la commande 4
; voice + $1B (1 octet): pour la commande 4
;
; CHIPx
; chip+$0 (1 long mot): adresse de
;         l'instrument
; chip+$4 (1 long mot): longueur restante
;         exprimée en octets
; chip+$8 (1 mot): volume actuel
; chip+$A (1 mot): période (=note)
; chip+$C (1 long mot): adresse de la par-
;         tie a répéter
; chip+$10 (1 long mot): longueur de la
;         boucle exprimée en octets
;
; Nous utilisons une table de volume au
; format suivant: seuls les 6 bits de poids
; faible sont utilisés dans chaque octet.
; Cela a pour conséquence d'avoir une très
; bonne qualité sonore.

; Soundtracker Playroutine

ST_CONNEXION
 BRA     INIT
 BRA     END
 BRA     PLAY
 ; Adresse de la musique
 DC.L 0
 ; Sortie désirée
 ; 0 = Soundchip
 ; 1 = ST Replay
 ; 2 = MV 16
 DC.W 0

INIT
 MOVEM.L D0-A6,-(SP)
 ; Adresse de la musique dans A0
 LEA     ST_CONNEXION(PC),A0
 MOVE.L  12(A0),A0
 LEA     OFFSET(PC),A1
 MOVE.W  #$1D8,POS(A1)
 MOVE.W  #$258,PATT(A1)
 MOVE.W  #$F,NBR(A1)
 ; Module du type 31 instruments?
 CMP.L   #'M.K.',$438(A0)
 BNE.S   NO_NEW_MODULES
 MOVE.W  #$3B8,POS(A1)
 MOVE.W  #$43C,PATT(A1)
 MOVE.W  #$1F,NBR(A1)

NO_NEW_MODULES
 LEA     ST_CONNEXION(PC),A0
 MOVE.L  12(A0),A0
 LEA     OFFSET(PC),A1
 ADD.W   POS(A1),A0
 MOVEQ   #$7F,D0
 MOVEQ   #0,D1

 ; Nombre de patterns
.LOOP1
 MOVE.L  D1,D2
 SUBQ.W  #1,D0

.LOOP2
 MOVE.B  (A0)+,D1
 CMP.B   D2,D1
 BGT.S   .LOOP1
 DBF     D0,.LOOP2
 ADDQ.B  #1,D2
 SWAP    D2
 LSR.L   #6,D2
 LEA     ST_CONNEXION(PC),A0
 MOVE.L  12(A0),A0
 ADD.L   A0,D2
 LEA     $14(A0),A0
 LEA     OFFSET(PC),A1
 MOVE.W  NBR(A1),D7
 ADD.W   PATT(A1),D2
 MOVE.L  D2,A2
 LEA     SAMPLESTARTS(PC),A1
 LEA     (A1),A3
 MOVEQ   #$1E,D6

.LOOP3
 CLR.L   (A3)+
 DBRA    D6,.LOOP3
 SUBQ.W  #1,D7

.LOOP4
 CLR.L   (A2)
 ; Adresse de début des Samples
 MOVE.L  A2,(A1)+
 MOVEQ   #0,D1
 MOVE.W  22(A0),D1
 LSL.L   #1,D1
 MOVE.L  A2,A5
 ADD.L   D1,A2
 MOVE.L  A2,(A1)
 LEA     (A2),A6
 MOVE.W  22(A0),D6
 BEQ.S   .NO_REVERSE
 SUBQ.W  #1,D6

 ; Inverse les Samples en mémoire
.LOOP5
 MOVE.B  (A5),D2
 MOVE.B  -1(A6),D3
 ; Met les Samples au format non signé
 EOR.B   #$80,D2
 EOR.B   #$80,D3
 MOVE.B  D3,(A5)+
 MOVE.B  D2,-(A6)
 DBRA    D6,.LOOP5

.NO_REVERSE
 CMP.W   #$0001,28(A0)
 BNE.S   .LOOP6
 CLR.W   28(A0)

.LOOP6
 LEA     $1E(A0),A0
 DBRA    D7,.LOOP4
 LEA     CHIP(PC),A0
 LEA     NO_INS(PC),A1
 MOVEQ   #$7,D0

.LOOP7
 MOVE.L  A1,(A0)+
 CLR.L   (A0)+
 CLR.L   (A0)+
 DBRA    D0,.LOOP7
 LEA     VOICES(PC),A0
 MOVEQ   #$1C-1,D0

.LOOP11
 CLR.L   (A0)+
 DBRA    D0,.LOOP11
 ; Initialise quelques Datas...
 LEA     OFFSET(PC),A0
 MOVE.B  #$6,SPEED(A0)
 CLR.B   SONGPOS(A0)
 CLR.B   COUNTER(A0)
 CLR.W   PATTPOS(A0)
 LEA     DIGIT_SWAP(PC),A0
 LEA     (DIGIT+1000)(PC),A1
 MOVE.L  A1,(A0)
 LEA     DIGIT(PC),A1
 MOVE.L  A1,4(A0)
 MOVE.W  #1000-1,D0

.LOOP8
 MOVE.W  #$400,(A1)+
 DBRA    D0,.LOOP8
 LEA     YET(PC),A0
 CMP.L   #'CNX',(A0)
 BEQ.S   .NO_CHANGE
 LEA     SOUND(PC),A0
 MOVE.W  #255,D0

.LOOP9
 MOVE.B  5(A0),6(A0)
 MOVE.B  5(A0),7(A0)
 MOVE.B  4(A0),5(A0)
 LEA     8(A0),A0
 DBRA    D0,.LOOP9
 LEA     VOLUME(PC),A0
 MOVE.W  #$3FFF,D0

.LOOP10
 MOVE.B  (A0),D1
 ; Transformation de la table de
 ; volume en données 6 bits
 LSR.B   #2,D1
 MOVE.B  D1,(A0)+
 DBRA    D0,.LOOP10
 LEA     YET(PC),A0
 MOVE.L  #'CNX',(A0)

.NO_CHANGE
 ; Yamaha reset
 MOVE.L  #$00000000,$FFFF8800.W
 MOVE.L  #$01010000,$FFFF8800.W
 MOVE.L  #$02020000,$FFFF8800.W
 MOVE.L  #$03030000,$FFFF8800.W
 MOVE.L  #$04040000,$FFFF8800.W
 MOVE.L  #$05050000,$FFFF8800.W
 MOVE.L  #$06060000,$FFFF8800.W
 MOVE.L  #$0707FFFF,$FFFF8800.W
 MOVE.L  #$08080000,$FFFF8800.W
 MOVE.L  #$09090000,$FFFF8800.W
 MOVE.L  #$0A0A0000,$FFFF8800.W
```

```
 ; Sauve les interruptions:
 LEA      STORE_INTERRUPTS(PC),A0
 ; VBL
 MOVE.L  $70.W,$0(A0)
 ; Timer A
 MOVE.L  $134.W,$4(A0)
 LEA      $FFFFFA00.W,A1
 ; IERA et IERB
 MOVEP.W $7(A1),D0
 MOVE.W  D0,$8(A0)
 ; IMRA IMRB VR et TACR
 MOVEP.L $13(A1),D0
 MOVE.L  D0,$A(A0)
 CLR.B   $19(A1)
 ; TADR
 MOVE.B  $1F(A1),$E(A0)
 ; Mouse OFF
 MOVE.B  #$12,$FFFFFC02.W
 ; Coupe le MFP
 CLR.B   $FFFFFA07.W
 CLR.B   $FFFFFA09.W
 ; Passage en fin automatique
 ; des interruptions
 BCLR.B  #$3,$FFFFFA17.W
 CLR.B   $FFFFFA19.W
 ; Positionne le TADR (registre
 ; de data du Timer A)
 MOVE.B  #FRQ_DATA,$FFFFFA1F.W
 ; Le prédiviseur divisé par 4
 MOVE.B  #1,$FFFFFA19.W
 ; Le Timer A est autorisé
 OR.B    #32,$FFFFFA07.W
 OR.B    #32,$FFFFFA13.W
 ; Mise en place du Timer A
 LEA      SOUNDCHIP(PC),A0
 MOVE.L  A0,$134.W
 LEA      ST_CONNEXION(PC),A1
 LEA      16(A1),A1
 CMP.W   #1,(A1)
 BNE.S   MV16?
 LEA      ST_REPLAY(PC),A0
 MOVE.L  A0,$134.W
MV16?
 CMP.W   #2,(A1)
 BNE.S   SOUNDCHIP?
 LEA      MV16(PC),A0
 MOVE.L  A0,$134.W
SOUNDCHIP?
 MOVEM.L (SP)+,D0-A6
 RTS

 ; Zone de sauvegarde du
 ; contexte du ST
STORE_INTERRUPTS
 REPT 8
 DC.W 0
 ENDR

END
 MOVEM.L D0-A6,-(SP)
 ; Restaure les interruptions:
 LEA      STORE_INTERRUPTS(PC),A0
 MOVE.L  $0(A0),$70.W
 MOVE.L  $4(A0),$134.W
 LEA      $FFFFFA00.W,A1
 MOVE.L  $A(A0),D0
 MOVEP.L D0,$13(A1)
 CLR.B   $19(A1)
 MOVE.B  $E(A0),$1F(A1)
 MOVE.B  $D(A0),$19(A1)
 MOVE.W  $8(A0),D0
 MOVEP.W D0,$7(A1)
```

```
 ; Mouse ON
 MOVE.B  #$8,$FFFFFC02.W
 MOVEM.L (SP)+,D0-A6
 RTS

PLAY
 MOVE.W  #$2700,SR
 MOVEM.L D0/A0,-(SP)
 ; Permutation des deux buffers
 ; de digit de travail
 LEA      DIGIT_SWAP(PC),A0
 MOVE.L  4(A0),D0
 MOVE.L  (A0),4(A0)
 MOVE.L  D0,(A0)
 LEA      GRRR(PC),A0
 MOVE.L  D0,2(A0)
 LEA      GRR(PC),A0
 MOVE.L  D0,2(A0)
 LEA      GR(PC),A0
 MOVE.L  D0,2(A0)
 MOVEM.L (SP)+,D0/A0
 MOVE.W  #$2300,SR
 ; Lit la partition et produit
 ; les effets sonores
 BSR      PLAYER
 ; Calcule la digit à jouer
 ; à la prochaine VBL
 BSR      PRECALC_DIGIT
 RTS

PLAYER
 MOVEM.L D0-A6,-(SP)
 LEA      OFFSET(PC),A4
 ADDQ.B  #$1,COUNTER(A4)
 MOVE.B  COUNTER(A4),D0
 ; Effets sonore ou nouvelle
 ; lecture de la partition?
 CMP.B   SPEED(A4),D0
 BLT.S   NONEW
 CLR.B   COUNTER(A4)
 BRA.S   READ_PATTERN

 ; Produit les effets sonores
 ; sur les 4 voies
 ; commandes:0,1,2,3,4,A
NONEW
 LEA      CHIP1(PC),A5
 LEA      VOICE1(PC),A6
 BSR      CHECK_COMMAND
 LEA      CHIP2(PC),A5
 LEA      VOICE2(PC),A6
 BSR      CHECK_COMMAND
 LEA      CHIP3(PC),A5
 LEA      VOICE3(PC),A6
 BSR      CHECK_COMMAND
 LEA      CHIP4(PC),A5
 LEA      VOICE4(PC),A6
 BSR      CHECK_COMMAND
 BRA      END_OF_PLAYER

 ; Lit la partition pour les 4 voies
READ_PATTERN
 LEA      ST_CONNEXION(PC),A0
 MOVE.L  12(A0),A0
 LEA      $C(A0),A3
 LEA      (A0),A2
 ADD.W   POS(A4),A2
 ADD.W   PATT(A4),A0

 MOVEQ   #0,D0
 MOVEQ   #0,D1
 MOVE.B  SONGPOS(A4),D0
```

```
 MOVE.B  0(A2,D0.W),D1
 SWAP     D1
 LSR.L   #$6,D1
 ADD.W   PATTPOS(A4),D1

 LEA      CHIP1(PC),A5
 LEA      VOICE1(PC),A6
 BSR.S   PLAYVOICE
 LEA      CHIP2(PC),A5
 LEA      VOICE2(PC),A6
 BSR.S   PLAYVOICE
 LEA      CHIP3(PC),A5
 LEA      VOICE3(PC),A6
 BSR.S   PLAYVOICE
 LEA      CHIP4(PC),A5
 LEA      VOICE4(PC),A6
 BSR.S   PLAYVOICE
 BRA      SET_REPEAT

 ; Routine qui traite une voie
PLAYVOICE
 MOVE.L  (A0,D1.L),(A6)
 ADDQ.L  #4,D1
 MOVEQ   #0,D2
 MOVE.B  $2(A6),D2
 AND.B   #$F0,D2
 LSR.B   #4,D2
 MOVE.B  (A6),D0
 AND.B   #$F0,D0
 OR.B    D0,D2
 TST.B   D2
 BEQ.S   SET_REGS
 MOVEQ   #0,D3
 LEA      SAMPLESTARTS(PC),A1
 MOVE.L  D2,D4
 SUBQ.L  #$1,D2
 ADD.W   D2,D2
 ADD.W   D2,D2
 MULU    #$1E,D4
 MOVE.L  (A1,D2.L),$4(A6)
 MOVE.W  (A3,D4.L),$8(A6)
 MOVE.W  $2(A3,D4.L),$12(A6)
 MOVE.W  $4(A3,D4.L),D3
 BRA.S   NO_LOOP
 MOVE.L  $4(A1,D2.L),$4(A6)
 ADD.W   $6(A3,D4.L),D3
 MOVE.W  D3,$8(A6)
 ADD.W   D3,D3
 SUB.L   D3,$4(A6)
 MOVE.L  $4(A6),$A(A6)
 MOVEQ   #0,D3
 MOVE.W  $6(A3,D4.L),D3
 MOVE.W  D3,$E(A6)
 MOVE.W  $12(A6),$8(A5)
 BRA.S   SET_REGS

NO_LOOP
 MOVE.L  $4(A6),$A(A6)
 MOVE.W  $6(A3,D4.L),$E(A6)
 MOVE.W  $12(A6),$8(A5)
SET_REGS
 MOVE.W  (A6),D0
 AND.W   #$FFF,D0
 BEQ      CHECK_COMMAND_2
 MOVE.B  $2(A6),D0
 AND.B   #$F,D0
 CMP.B   #$3,D0
 BNE.S   SETPERIOD
 BSR      SET_PORTAMENTO_TONE
 BRA      CHECK_COMMAND_2
SETPERIOD
 MOVE.W  (A6),$10(A6)
```

```
AND.W   #$FFF,$10(A6)
CLR.B   $1B(A6)

MOVE.L  $4(A6),(A5)
MOVEQ   #0,D0
MOVE.W  $8(A6),D0
ADD.W   D0,D0
MOVE.L  D0,$4(A5)
MOVE.W  $10(A6),$A(A5)
BRA     CHECK_COMMAND_2

; Active les repeat des 4 voies
SET_REPEAT
MOVEQ   #0,D0
LEA     CHIP1(PC),A5
LEA     VOICE1(PC),A6
MOVE.L  $A(A6),$C(A5)
MOVE.W  $E(A6),D0
ADD.W   D0,D0
MOVE.L  D0,$10(A5)
LEA     CHIP2(PC),A5
LEA     VOICE2(PC),A6
MOVE.L  $A(A6),$C(A5)
MOVE.W  $E(A6),D0
ADD.W   D0,D0
MOVE.L  D0,$10(A5)
LEA     CHIP3(PC),A5
LEA     VOICE3(PC),A6
MOVE.L  $A(A6),$C(A5)
MOVE.W  $E(A6),D0
ADD.W   D0,D0
MOVE.L  D0,$10(A5)
LEA     CHIP4(PC),A5
LEA     VOICE4(PC),A6
MOVE.L  $A(A6),$C(A5)
MOVE.W  $E(A6),D0
ADD.W   D0,D0
MOVE.L  D0,$10(A5)

ADD.W   #$10,PATTPOS(A4)
; Nouveau pattern?
CMP.W   #$400,PATTPOS(A4)
BNE.S   END_OF_PLAYER
NEXT_PATT
CLR.W   PATTPOS(A4)
CLR.B   BREAK(A4)
ADDQ.B  #1,SONGPOS(A4)
AND.B   #$7F,SONGPOS(A4)
MOVE.B  SONGPOS(A4),d1
LEA     ST_CONNEXION(PC),A0
MOVE.L  12(A0),A0
ADD.W   POS(A4),A0
CMP.B   -2(A0),d1
BNE.S   END_OF_PLAYER
CLR.B   SONGPOS(A4)

END_OF_PLAYER
TST.B   BREAK(A4)
BNE.S   NEXT_PATT
MOVEM.L (SP)+,D0-A6
RTS

; Effets sonores qui se
; produisent lorsqu'il ne
; faut pas lire la partition
CHECK_COMMAND
MOVE.W  $2(A6),D0
AND.W   #$FFF,D0
; Pas de commande
BEQ.S   NO_COMMAND
MOVE.B  $2(A6),D0
AND.B   #$F,D0
BEQ.S   ARPEGGIO
CMP.B   #$1,D0
BEQ     PORTAMENTO_UP
CMP.B   #$2,D0
BEQ     PORTAMENTO_DOWN
CMP.B   #$3,D0
BEQ     PORTAMENTO_TONE
CMP.B   #$4,D0
BEQ     VIBRATO
MOVE.W  $10(A6),$A(A5)
CMP.B   #$A,D0
BEQ     VOLUME_SLIDE
NO_COMMAND
RTS

; Commande 0: ARPEGGIO
ARPEGGIO
MOVEQ   #0,D0
MOVE.B  COUNTER(A4),D0
DIVS    #$3,D0
SWAP    D0
CMP.W   #$0,D0
BEQ.S   ARPEGGIO2
CMP.W   #$2,D0
BEQ.S   ARPEGGIO1

MOVEQ   #0,D0
MOVE.B  $3(A6),D0
LSR.B   #$4,D0
BRA.S   ARPEGGIO3
ARPEGGIO1
MOVEQ   #0,D0
MOVE.B  $3(A6),D0
AND.B   #$F,D0
BRA.S   ARPEGGIO3
ARPEGGIO2
MOVE.W  $10(A6),D2
BRA.S   ARPEGGIO4
ARPEGGIO3
ADD.W   D0,D0
MOVEQ   #0,D1
MOVE.W  $10(A6),D1
LEA     PERIODS(PC),A0
MOVEQ   #$24,D7
ARPEGGIO_LOOP
MOVE.W  (A0,D0.W),D2
CMP.W   (A0),D1
BGE.S   ARPEGGIO4
LEA     2(A0),A0
DBRA    D7,ARPEGGIO_LOOP
RTS
ARPEGGIO4
MOVE.W  D2,$A(A5)
RTS

; Commande 1: PORTAMENTO UP
PORTAMENTO_UP
MOVEQ   #0,D0
MOVE.B  $3(A6),D0
SUB.W   D0,$10(A6)
MOVE.W  $10(A6),D0
CMP.W   #$71,D0
BPL.S   PORTUP
MOVE.W  #$71,$10(A6)
PORTUP
MOVE.W  $10(A6),$A(A5)
RTS

; Commande 2: PORTAMENTO DOWN
PORTAMENTO_DOWN
MOVEQ   #0,D0
MOVE.B  $3(A6),D0
ADD.W   D0,$10(A6)
MOVE.W  $10(A6),D0
CMP.W   #$358,D0
BMI.S   PORTDWN
MOVE.W  #$358,$10(A6)
PORTDWN
MOVE.W  $10(A6),$A(A5)
RTS

; Met en place le
; portamento tone
SET_PORTAMENTO_TONE
MOVE.W  (A6),D2
AND.W   #$FFF,D2
MOVE.W  D2,$18(A6)
MOVE.W  $10(A6),D0
CLR.B   $16(A6)
CMP.W   D0,D2
BEQ.S   CLR_PORTAMENTO_TONE
BGE.S   OK_RTS
MOVE.B  #$1,$16(A6)
RTS
CLR_PORTAMENTO_TONE
CLR.W   $18(A6)
OK_RTS
RTS

; Commande 3: PORTAMENTO TONE
PORTAMENTO_TONE
MOVE.B  $3(A6),D0
BEQ.S   MY_SLIDE
MOVE.B  D0,$17(A6)
CLR.B   $3(A6)
MY_SLIDE
TST.W   $18(A6)
BEQ.S   OK_RTS
MOVEQ   #0,D0
MOVE.B  $17(A6),D0
TST.B   $16(A6)
BNE.S   MY_SUB
ADD.W   D0,$10(A6)
MOVE.W  $18(A6),D0
CMP.W   $10(A6),D0
BGT.S   MY_OK
MOVE.W  $18(A6),$10(A6)
CLR.W   $18(A6)
MY_OK
MOVE.W  $10(A6),$A(A5)
RTS
MY_SUB
SUB.W   D0,$10(A6)
MOVE.W  $18(A6),D0
CMP.W   $10(A6),D0
BLT.S   MY_OK
MOVE.W  $18(A6),$10(A6)
CLR.W   $18(A6)
MOVE.W  $10(A6),$A(A5)
RTS

; Commande 4: VIBRATO
VIBRATO
MOVE.B  $3(A6),D0
BEQ.S   VIB
MOVE.B  D0,$1A(A6)

VIB
MOVE.B  $1B(A6),D0
LEA     SINUS(PC),A0
LSR.W   #$2,D0
AND.W   #$1F,D0
MOVEQ   #0,D2
MOVE.B  (A0,D0.W),D2
```

```
 MOVE.B   $1A(A6),D0
 AND.W    #$F,D0
 MULU     D0,D2
 LSR.W    #$6,D2
 MOVE.W   $10(A6),D0
 TST.B    $1B(A6)
 BMI.S    VIB_MIN
 ADD.W    D2,D0
 BRA.S    VIB2
VIB_MIN
 SUB.W    D2,D0
VIB2
 MOVE.W   D0,$A(A5)
 MOVE.B   $1A(A6),D0
 LSR.W    #$2,D0
 AND.W    #$3C,D0
 ADD.B    D0,$1B(A6)
 RTS

 ; Commande $A: VOLUME SLIDE
VOLUME_SLIDE
 MOVEQ    #0,D0
 MOVE.B   $3(A6),D0
 LSR.B    #4,D0
 TST.B    D0
 BEQ.S    VOLUME_DOWN
 ADD.W    D0,$12(A6)
 CMP.W    #$40,$12(A6)
 BMI.S    VOLUP_OK
 MOVE.W   #$40,$12(A6)
VOLUP_OK
 MOVE.W   $12(A6),$8(A5)
 RTS

VOLUME_DOWN
 MOVEQ    #0,D0
 MOVE.B   $3(A6),D0
 AND.B    #$F,D0
 SUB.W    D0,$12(A6)
 BPL.S    VOLDWN_OK
 CLR.W    $12(A6)
VOLDWN_OK
 MOVE.W   $12(A6),$8(A5)
 RTS

 ; Commandes qui peuvent
 ; être effectuées lors de
 ; la lecture de la partition
CHECK_COMMAND_2
 MOVE.B   $2(A6),D0
 AND.B    #$F,D0
 CMP.B    #$B,D0
 BEQ.S    POSITION_JUMP
 CMP.B    #$C,D0
 BEQ.S    SET_VOLUME
 CMP.B    #$D,D0
 BEQ.S    PATTERN_BREAK
 CMP.B    #$E,D0
 BEQ.S    SET_FILTER
 CMP.B    #$F,D0
 BEQ.S    SET_SPEED
 ; Pas de commande
 RTS

 ; Commande $B: POSITION JUMP
POSITION_JUMP
 MOVE.B   $3(A6),D0
 SUBQ.B   #$1,D0
 MOVE.B   D0,SONGPOS(A4)
 NOT.B    BREAK(A4)
 RTS
```

```
 ; Commande $C: SET VOLUME
SET_VOLUME
 CMP.B    #$40,$3(A6)
 BLE.S    VOLUME_OK
 MOVE.B   #$40,$3(A6)
VOLUME_OK
 MOVE.B   $3(A6),$9(A5)
 RTS

 ; Commande $D: PATTERN BREAK
PATTERN_BREAK
 NOT.B    BREAK(A4)
 RTS

 ; Commande $E: FILTER
 ; (il n'est pas implémenté ici)
SET_FILTER
 RTS

 ; commande $F: SET SPEED
SET_SPEED
 MOVE.B   $3(A6),D0
 AND.W    #$1F,D0
 BEQ.S    NO_CHANGE_SPEED
 CLR.B    COUNTER(A4)
 MOVE.B   D0,SPEED(A4)
NO_CHANGE_SPEED
 RTS

 ; Précalcule la digit qui
 ; sera jouée lors de la
 ; VBL suivante
PRECALC_DIGIT
 MOVEM.L  D0-A6,-(SP)
 LEA      VOLUME(PC),A3
 LEA      FRQ(PC),A4
 LEA      MEGABUFFER(PC),A5
 LEA      CHIP1(PC),A6
 BSR.S    CALC_VOICE
 LEA      MEGABUFFER+500(PC),A5
 LEA      CHIP2(PC),A6
 BSR.S    CALC_VOICE
 LEA      MEGABUFFER+1000(PC),A5
 LEA      CHIP3(PC),A6
 BSR.S    CALC_VOICE
 LEA      MEGABUFFER+1500(PC),A5
 LEA      CHIP4(PC),A6
 BSR.S    CALC_VOICE

 LEA      MEGABUFFER(PC),A0
 LEA      500(A0),A1
 LEA      500(A1),A2
 LEA      500(A2),A3
 LEA      DIGIT_SWAP(PC),A4
 MOVE.L   4(A4),A4
 BSR      MIX_ALL_VOICES
 MOVEM.L  (SP)+,D0-A6
 RTS

 ; Calcul de la digit
 ; pour une voie
CALC_VOICE
 ; Adresse instrument
 MOVE.L   $0(A6),A0
 ; Valeur décimale du pointeur
 MOVEQ    #0,D1
 ; Longueur du sample
 MOVE.L   $4(A6),D3
 ; Volume
 MOVE.W   $8(A6),D4
 BEQ.S    .VOLUME_NUL
 SUBQ.W   #1,D4
```

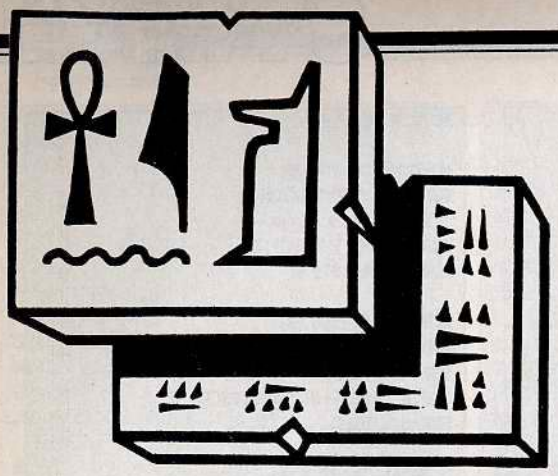
```
.VOLUME_NUL
 LSL.W    #$8,D4
 ; Repeat lenght
 MOVE.L   $10(A6),D6
 ; Période
 MOVE.W   $A(A6),D7
 ; *2
 ADD.W    D7,D7
 ; *2
 ADD.W    D7,D7
 ; Période.L
 MOVE.L   (A4,D7.W),D7
 ; .W bas de la période
 MOVE.W   D7,D0
 SWAP     D7
 MOVEQ    #0,D2
 ; .W haut de la période
 MOVE.W   D7,D2
 SWAP     D7
 ; Repeat?
 TST.L    D6
 BNE.S    .0
 ; Pas repeat -> frq=0
 CLR.L    D7
 ; Adresse loop
.0
 MOVE.L   $C(A6),A2

 MOVEQ    #FRQ_LOOP,D5
.PAULA
 ; Partie décimale de la fréquence
 SUB.W    D0,D1
 ; Partie entière de la fréquence
 ; + retenue (s'il y a lieu)
 SUBX.W   D2,D3
 ; Fin du sample?
 BCS.S    .R1
.RR1
 ; Charge le sample
 MOVE.B   0(A0,D3.L),D4
 ; Charge sa correspondance avec le volume
 MOVE.B   0(A3,D4.W),(A5)+
 ; Et on recommence...
 SUB.W    D0,D1
 SUBX.W   D2,D3
 BCS.S    .R2
.RR2
 MOVE.B   0(A0,D3.L),D4
 MOVE.B   0(A3,D4.W),(A5)+
 SUB.W    D0,D1
 SUBX.W   D2,D3
 BCS.S    .R3
.RR3
 MOVE.B   0(A0,D3.L),D4
 MOVE.B   0(A3,D4.W),(A5)+
 ; La même routine est exécutee 3 fois
 ; cela permet d'eviter 2 dbra
 DBRA     D5,.PAULA
 BRA.S    END_OF_CALC_A_VOICE

.R1
 MOVE.L   D6,D3
 MOVE.W   D7,D0
 SWAP     D7
 MOVE.W   D7,D2
 SWAP     D7
 MOVE.L   A2,A0
 BRA.S    .RR1
.R2
 MOVE.L   D6,D3
 MOVE.W   D7,D0
 SWAP     D7
```

```
 MOVE.W  D7,D2
 SWAP    D7
 MOVE.L  A2,A0
 BRA.S   .RR2
.R3
 MOVE.L  D6,D3
 MOVE.W  D7,D0
 SWAP    D7
 MOVE.W  D7,D2
 SWAP    D7
 MOVE.L  A2,A0
 BRA.S   .RR3
END_OF_CALC_A_VOICE
 ; Adresse du sample à sauver
 MOVE.L  A0,$0(A6)
 ; Position dans le sample à sauver
 MOVE.L  D3,$4(A6)
 RTS

 ; Mixe les 4 voies
MIX_ALL_VOICES
 MOVEQ   #FRQ_LOOP,D1
.LOOP
 MOVEQ   #0,D0
 ; V1
 MOVE.B  (A0)+,D0
 ; V1+V2
 ADD.B   (A1)+,D0
 ; V1+V2+V3
 ADD.B   (A2)+,D0
 ; V1+V2+V3+V4
 ADD.B   (A3)+,D0
 ; Multiplie par 8
 LSL.W   #3,D0
 ; Sauve le résultat pour être exécuté
 ; la VBL d'après
 MOVE.W  D0,(A4)+
 MOVEQ   #0,D0
 MOVE.B  (A0)+,D0
 ADD.B   (A1)+,D0
 ADD.B   (A2)+,D0
 ADD.B   (A3)+,D0
 LSL.W   #3,D0
 MOVE.W  D0,(A4)+
 MOVEQ   #0,D0
 MOVE.B  (A0)+,D0
 ADD.B   (A1)+,D0
 ADD.B   (A2)+,D0
 ADD.B   (A3)+,D0
 LSL.W   #3,D0
 MOVE.W  D0,(A4)+
 ; La même routine est exécutée 3 fois
 ; cela permet d'eviter 2 DBRA
 DBRA    D1,.LOOP
 MOVE.W  D0,(A4)+
 MOVE.W  D0,(A4)+
 MOVE.W  D0,(A4)+
 RTS

 ; Gestion cartouche MV16
MV16
 MOVEM.L D0/A0,-(SP)
GR
 LEA     DIGIT,A0
 ; Lit le mot de digit
 MOVE.W  (A0),D0
 LEA     (GR+4)(PC),A0
 ; Incrémente adresse du buffer
 ADD.W   #2,(A0)
 LEA     $FFFA0000,A0
 ; Format octet pour la cartouche
 ; du type: %1111111111110
 ; 13 bits avec les 12 bits de poids
 ; fort de significatif
 ADD.W   D0,D0
 ADD.W   D0,D0
 MOVE.B  0(A0,D0.W),D0
 MOVE.L  (SP)+,D0
 MOVE.L  (SP)+,A0
 RTE

 ; Gestion cartouche ST Replay
ST_REPLAY
 MOVEM.L D0/A0,-(SP)
GRR
 LEA     DIGIT,A0
 ; Lit le mot de digit
 MOVE.W  (A0),D0
 LEA     (GRR+4)(PC),A0
 ; Incrémente adresse du buffer
 ADD.W   #2,(A0)
 LEA     $FFFA0000,A0
 ; Format octet pour la cartouche
 ; du type: %111111110
 ; 9 bits avec les 8 bits de poids
 ; fort de significatif
 LSR.W   #2,D0
 MOVE.B  0(A0,D0.W),D0
 MOVE.L  (SP)+,D0
 MOVE.L  (SP)+,A0
 RTE

 ; Gestion Yamaha 2149
SOUNDCHIP
 MOVEM.L D0/A0,-(SP)
GRRR
 LEA     DIGIT,A0
 ; Lit le mot de digit
 MOVE.W  (A0),D0
 LEA     (GRRR+4)(PC),A0
 ; Incrémente adresse du buffer
 ADD.W   #2,(A0)
 ; Pointe sur la valeur de la table
 ; ST Replay pour le Soundchip
 LEA     SOUND(PC,D0.W),A0
 MOVE.L  (A0)+,D0
 MOVE.L  (A0),$FFFF8800.W
 LEA     $FFFF8800.W,A0
 MOVEP.L D0,(A0)
 MOVE.L  (SP)+,D0
 MOVE.L  (SP)+,A0
 RTE

 ; Table ST Replay
 ; COPYRIGHT 2-BIT SYSTEM / MICRODEAL

SOUND
 DC.W $80E,$90D,$A0C,0,$80F,$903,$A00,0
 DC.W $80F,$903,$A00,0,$80F,$903,$A00,0
 DC.W $80F,$903,$A00,0,$80F,$903,$A00,0
 DC.W $80F,$903,$A00,0,$80E,$90D,$A0B,0
 DC.W $80E,$90D,$A0B,0,$80E,$90D,$A0B,0
 DC.W $80E,$90D,$A0B,0,$80E,$90D,$A0B,0
 DC.W $80E,$90D,$A0B,0,$80E,$90D,$A0B,0
 DC.W $80E,$90D,$A0A,0,$80E,$90D,$A0A,0
 DC.W $80E,$90D,$A0A,0,$80E,$90D,$A0A,0
 DC.W $80E,$90C,$A0C,0,$80E,$90D,$A00,0
 DC.W $80D,$90D,$A0D,0,$80D,$90D,$A0D,0
 DC.W $80D,$90D,$A0D,0,$80D,$90D,$A0D,0
 DC.W $80D,$90D,$A0D,0,$80D,$90D,$A0D,0
 DC.W $80E,$90C,$A0B,0,$80E,$90C,$A0B,0
 DC.W $80E,$90C,$A0B,0,$80E,$90C,$A0B,0
 DC.W $80E,$90C,$A0B,0,$80E,$90C,$A0B,0
 DC.W $80E,$90C,$A0B,0,$80E,$90C,$A0B,0
 DC.W $80E,$90C,$A0A,0,$80E,$90C,$A0A,0
 DC.W $80E,$90C,$A0A,0,$80E,$90C,$A0A,0
 DC.W $80D,$90D,$A0C,0,$80D,$90D,$A0C,0
 DC.W $80E,$90C,$A09,0,$80E,$90C,$A09,0
 DC.W $80E,$90C,$A05,0,$80E,$90C,$A00,0
 DC.W $80E,$90C,$A00,0,$80E,$90B,$A0B,0
 DC.W $80E,$90B,$A0B,0,$80E,$90B,$A0B,0
 DC.W $80E,$90B,$A0B,0,$80E,$90B,$A0A,0
 DC.W $80E,$90B,$A0A,0,$80E,$90B,$A0A,0
 DC.W $80D,$90D,$A0B,0,$80D,$90D,$A0B,0
 DC.W $80D,$90D,$A0B,0,$80E,$90B,$A09,0
 DC.W $80E,$90B,$A09,0,$80E,$90B,$A09,0
 DC.W $80D,$90C,$A0C,0,$80D,$90D,$A0A,0
 DC.W $80E,$90B,$A07,0,$80E,$90B,$A00,0
 DC.W $80E,$90B,$A00,0,$80D,$90D,$A09,0
 DC.W $80D,$90D,$A09,0,$80E,$90A,$A09,0
 DC.W $80D,$90D,$A08,0,$80D,$90D,$A07,0
 DC.W $80D,$90D,$A04,0,$80D,$90D,$A00,0
 DC.W $80E,$90A,$A04,0,$80E,$909,$A09,0
 DC.W $80E,$909,$A09,0,$80D,$90C,$A0B,0
 DC.W $80E,$909,$A08,0,$80E,$909,$A08,0
 DC.W $80E,$909,$A07,0,$80E,$908,$A08,0
 DC.W $80E,$909,$A01,0,$80C,$90C,$A0C,0
 DC.W $80D,$90C,$A0A,0,$80E,$908,$A06,0
 DC.W $80E,$907,$A07,0,$80E,$908,$A00,0
 DC.W $80E,$907,$A05,0,$80E,$906,$A06,0
 DC.W $80D,$90C,$A09,0,$80E,$905,$A05,0
 DC.W $80E,$904,$A04,0,$80D,$90C,$A08,0
 DC.W $80D,$90B,$A0B,0,$80E,$900,$A00,0
 DC.W $80D,$90C,$A06,0,$80D,$90C,$A05,0
 DC.W $80D,$90C,$A02,0,$80C,$90C,$A0B,0
 DC.W $80C,$90C,$A0B,0,$80D,$90B,$A0A,0
 DC.W $80D,$90B,$A0A,0,$80D,$90B,$A0A,0
 DC.W $80D,$90B,$A0A,0,$80C,$90C,$A0A,0
 DC.W $80C,$90C,$A0A,0,$80C,$90C,$A0A,0
 DC.W $80D,$90B,$A09,0,$80D,$90B,$A09,0
 DC.W $80D,$90A,$A0A,0,$80D,$90A,$A0A,0
 DC.W $80D,$90A,$A0A,0,$80C,$90C,$A09,0
 DC.W $80C,$90C,$A09,0,$80C,$90C,$A09,0
 DC.W $80D,$90B,$A06,0,$80C,$90B,$A0B,0
 DC.W $80C,$90C,$A08,0,$80D,$90B,$A00,0
 DC.W $80D,$90B,$A00,0,$80C,$90C,$A07,0
 DC.W $80C,$90C,$A06,0,$80C,$90C,$A05,0
 DC.W $80C,$90C,$A03,0,$80C,$90C,$A01,0
 DC.W $80C,$90B,$A0A,0,$80D,$90A,$A05,0
 DC.W $80D,$90A,$A04,0,$80D,$90A,$A02,0
 DC.W $80D,$909,$A08,0,$80D,$909,$A08,0
 DC.W $80C,$90B,$A09,0,$80C,$90B,$A09,0
 DC.W $80D,$908,$A08,0,$80B,$90B,$A0B,0
 DC.W $80D,$909,$A05,0,$80C,$90B,$A08,0
 DC.W $80D,$909,$A02,0,$80D,$908,$A06,0
 DC.W $80C,$90B,$A07,0,$80D,$907,$A07,0
 DC.W $80C,$90B,$A06,0,$80C,$90A,$A09,0
 DC.W $80B,$90B,$A0A,0,$80C,$90B,$A02,0
 DC.W $80C,$90B,$A00,0,$80C,$90A,$A08,0
 DC.W $80D,$906,$A04,0,$80D,$905,$A05,0
 DC.W $80D,$905,$A04,0,$80C,$909,$A09,0
 DC.W $80D,$904,$A03,0,$80B,$90B,$A09,0
 DC.W $80C,$90A,$A05,0,$80B,$90A,$A0A,0
 DC.W $80C,$909,$A08,0,$80B,$90B,$A08,0
 DC.W $80C,$90A,$A00,0,$80C,$90A,$A00,0
 DC.W $80C,$909,$A07,0,$80B,$90B,$A07,0
 DC.W $80C,$909,$A06,0,$80B,$90B,$A06,0
 DC.W $80B,$90A,$A09,0,$80B,$90B,$A05,0
 DC.W $80A,$90A,$A0A,0,$80B,$90B,$A02,0
 DC.W $80B,$90A,$A08,0,$80C,$907,$A07,0
 DC.W $80C,$908,$A04,0,$80C,$907,$A06,0
 DC.W $80B,$909,$A09,0,$80C,$906,$A06,0
 DC.W $80A,$90A,$A09,0,$80C,$907,$A03,0
 DC.W $80B,$90A,$A05,0,$80B,$909,$A08,0
 DC.W $80B,$90A,$A03,0,$80A,$90A,$A08,0
 DC.W $80B,$90A,$A00,0,$80B,$909,$A07,0
```

```
 DC.W $80B,$908,$A08,0,$80A,$90A,$A07,0
 DC.W $80A,$909,$A09,0,$80C,$901,$A01,0
 DC.W $80A,$90A,$A06,0,$80B,$908,$A07,0
 DC.W $80A,$90A,$A05,0,$80A,$909,$A08,0
 DC.W $80A,$90A,$A02,0,$80A,$90A,$A01,0
 DC.W $80A,$90A,$A00,0,$809,$909,$A09,0
 DC.W $80A,$908,$A08,0,$80B,$908,$A01,0
 DC.W $80A,$909,$A06,0,$80B,$907,$A04,0
 DC.W $80A,$909,$A05,0,$809,$909,$A08,0
 DC.W $80A,$909,$A03,0,$80A,$908,$A06,0
 DC.W $80A,$909,$A00,0,$809,$909,$A07,0
 DC.W $809,$908,$A08,0,$80A,$908,$A04,0
 DC.W $809,$909,$A06,0,$80A,$908,$A01,0
 DC.W $809,$909,$A05,0,$809,$908,$A07,0
 DC.W $808,$908,$A08,0,$809,$909,$A02,0
 DC.W $809,$908,$A06,0,$809,$909,$A00,0
 DC.W $809,$907,$A07,0,$808,$908,$A07,0
 DC.W $809,$907,$A06,0,$809,$908,$A02,0
 DC.W $808,$908,$A06,0,$809,$906,$A06,0
 DC.W $808,$907,$A07,0,$808,$908,$A04,0
 DC.W $808,$907,$A06,0,$808,$908,$A02,0
 DC.W $807,$907,$A07,0,$808,$906,$A06,0
 DC.W $808,$907,$A04,0,$807,$907,$A06,0
 DC.W $808,$906,$A05,0,$808,$906,$A04,0
 DC.W $807,$906,$A06,0,$807,$907,$A04,0
 DC.W $808,$905,$A04,0,$806,$906,$A06,0
 DC.W $807,$906,$A04,0,$807,$905,$A05,0
 DC.W $806,$906,$A05,0,$806,$906,$A04,0
 DC.W $806,$905,$A05,0,$806,$906,$A02,0
 DC.W $806,$905,$A04,0,$805,$905,$A05,0
 DC.W $806,$905,$A02,0,$805,$905,$A04,0
 DC.W $805,$904,$A04,0,$805,$905,$A02,0
 DC.W $804,$904,$A04,0,$804,$904,$A03,0
 DC.W $804,$904,$A02,0,$804,$903,$A03,0
 DC.W $803,$903,$A03,0,$803,$903,$A02,0
 DC.W $803,$902,$A02,0,$802,$902,$A02,0
 DC.W $802,$902,$A01,0,$801,$901,$A01,0
 DC.W $802,$901,$A00,0,$801,$901,$A00,0
 DC.W $801,$900,$A00,0,$800,$900,$A00,0

 ; Table pour le vibrato
SINUS
 DC.B $00,$18,$31,$4A,$61,$78,$8D,$A1,$B4
 DC.B $C5,$D4,$E0,$EB,$F4,$FA,$FD
 DC.B $FF,$FD,$FA,$F4,$EB,$E0,$D4,$C5,$B4
 DC.B $A1,$8D,$78,$61,$4A,$31,$18

 ; Table pour l'arpeggio
PERIODS
 dc.w $0358,$0328,$02FA,$02D0,$02A6,$0280
 dc.w $025C,$023A,$021A,$01FC,$01E0
 dc.w $01C5,$01AC,$0194,$017D,$0168,$0153
 dc.w $0140,$012E,$011D,$010D,$00FE
 dc.w $00F0,$00E2,$00D6,$00CA,$00BE,$00B4
 dc.w $00AA,$00A0,$0097,$008F,$0087
 dc.w $007F,$0078,$0071,$0000,$0000

OFFSET
 REPT $6
 DC.W 0
 ENDR
SPEED     EQU $0
SONGPOS   EQU $1
PATTPOS   EQU $2
COUNTER   EQU $4
BREAK     EQU $5
POS       EQU $6
PATT      EQU $8
NBR       EQU $A
SAMPLESTARTS
 REPT $20
 DC.L 0
 ENDR
VOICES
 REPT $1C
 DC.L 0
 ENDR
VOICE1    EQU VOICES
VOICE2    EQU VOICES+$1C
VOICE3    EQU VOICES+$38
VOICE4    EQU VOICES+$54
CHIP
 REPT $18
 DC.L 0
 ENDR
CHIP1     EQU CHIP
CHIP2     EQU CHIP+$18
CHIP3     EQU CHIP+$30
CHIP4     EQU CHIP+$48
NO_INS    DC.W $8080
MEGABUFFER
 REPT 500
 DC.L 0
 ENDR
DIGIT
 REPT 500*2
 DC.W 0
 ENDR
DIGIT_SWAP
 REPT 2
 DC.L 0
 ENDR
FRQ_LOOP EQU  92
FRQ_DATA EQU  44

 ; Voici les valeurs pour changer
 ; de fréquence...
 ; Fréquences: 14Khz  12.3Khz  10.1Khz
 ; FRQ_DATA    44      50       61
 ; FRQ_LOOP    92      81       66
 ; Fréquences: 8.5Khz  7.5Khz
 ; FRQ_DATA    72      82
 ; FRQ_LOOP    56      49

FRQ       DC.L 0
          INCBIN '14_KHZ.FRQ'
YET       DC.L 0
VOLUME    INCBIN 'VOLUME.BIN'
```

```
; Routine d'exemple utilisant la routine
; de replay de musiques Soundtracker

 ; Supexec
 PEA      MAIN(PC)
 MOVE.W   #$26,-(SP)
 TRAP     #14
 LEA      6(SP),SP

 ; Exit
 CLR.W    -(SP)
 TRAP     #1

 ; Programme exécuté en Superviseur
MAIN
 ; Init la musique et ses paramètres
 LEA      STCONNEXION(PC),A0
 ; Poke l'adresse de la musique
 LEA      MUSIC(PC),A1
 MOVE.L   A1,12(A0)
 ; Choix de la sortie...
 ; Sounchip = 0
 ; ST Replay = 1
 ; MV16 = 2
 MOVE.W   #0,16(A0)

 MOVE.W   #$2700,SR
 BSR      STCONNEXION
 ; Lance la musique
 LEA      MY_VBL(PC),A0
 MOVE.L   A0,$70.W

 MOVE.W   #$2300,SR
DEMO
 BSR      VSYNC
 CMP.B    #$39+$80,$FFFFFC02.W
 BNE.S    DEMO

 ; Remet le ST dans l'état où
 ; nous l'avons trouvé
 MOVE.W   #$2700,SR
 BSR      STCONNEXION+$4
 MOVE.W   #$2300,SR
 ; Wait MFP
 MOVE.W   #4000,D0
WAIT_MFP
 NOP
 DBRA     D0,WAIT_MFP
 ; Supexec fin
 RTS

MY_VBL
 MOVE.L   A0,-(SP)
 LEA      SYNC(PC),A0
 ST       (A0)
 MOVE.L   (SP)+,A0
 BSR      STCONNEXION+$8
 RTE

VSYNC
 MOVE.L   A0,-(SP)
 LEA      SYNC(PC),A0
 SF       (A0)
WAITVBL
 TST.B    (A0)
 BEQ.S    WAITVBL
 MOVE.L   (SP)+,A0
 RTS

SYNC DS.W 1

 ; On inclut ici le fichier
 ; contenant le source de la
 ; playroutine...
STCONNEXION
 INCLUDE 'ST_MAG.S'

 ; Et ici on inclut la musique
 ; soundtrack.
MUSIC
 INCBIN MUZIK.MOD
```

3615
STMAG



DES LUTINS ET DES SOURIS

Quoi de plus agréable que de personnaliser ses propres programmes avec des petits lutins sur mesure ou une souris en forme de flèche indienne. Souvent, on renonce à cette possibilité à cause de la difficulté de mise en oeuvre de la définition binaire, sous forme de chaînes de caractères souvent barbares, de ces bestioles. Ce programme d'édition de lutins et de souris vous permet de gérer facilement, et au pixel près, les motifs de ces derniers. Tout se fait à la souris et quant au dessin, les fonctions élémentaires (inversion, copie vers le masque, sauvegarde, et patati, et patata...) sont gérées au clavier afin d'éviter une routine d'interrogation souris qui aurait pris un grand nombre de lignes.

Au lancement deux grilles s'affichent : à gauche, celle du corps, et à droite, celle du masque, le résultat est affiché dans les petits cadres en haut de l'écran. Il suffit de cliquer avec le bouton gauche de la souris pour valider un pixel et avec le bouton droit pour l'effacer. Les flèches de déplacement permettent de déplacer le point d'action qui se distingue par un motif spécial. Un format particulier (.LUT) permet de conserver une trace en vue d'une édition ultérieure.

Franck-Olivier LELAIDIER

```
REM ************************
REM *  EDITEUR DE SPRITES  *
REM *  TOUTES RESOLUTIONS  *
REM ************************
DEFWRD "a-z"
OPTION BASE 1
resolution
init
init_disquette
init_fleche
init_lutin(1,1,0,0,15,lutin1$)
init_lutin(1,1,0,15,15,lutin2$)
DEFFILL vert,2,2
PBOX 0,0,320*cx,200*cy
effacer
forme_souris(9)
cadre_selection
DO
  MOUSE xs,ys,es
  IF es=1 OR es=2
    choix_grille(dx1,dy1,xs,ys,noir)
    choix_grille(dx2,dy2,xs,ys,rouge)
  ELSE
    rep=CVI(INKEY$)-58
    SELECT rep
    CASE 1
      inversion
      affiche_grille
    CASE 2
      copie_masque
      affiche_grille
    CASE 3
```

```
      sauve_objet(1)
    CASE 4
      sauve_objet(2)
    CASE 5
      sauve_objet(3)
    CASE 6
      charge_objet
    CASE 7
      ALERT 2,"|VOULEZ-VOUS EFFACER L'OBJET ?",1,"NON|OUI",choix
      IF choix=2
        efface_objet
      ENDIF
    CASE 8,9
      ALERT 1,"|FONCTION INEXISTANTE|A VOUS DE JOUER !",1,"OK",choix
    CASE 10
      ALERT 2,"|VOULEZ-VOUS VRAIMENT QUITTER ?",1,"NON|OUI",choix
      IF choix=2
        END
      ENDIF
    CASE 14,22,17,19
      pxai=pxa
      pyai=pya
      SELECT rep
      CASE 14
        DEC pya
      CASE 17
        DEC pxa
      CASE 19
        INC pxa
      CASE 22
        INC pya
      ENDSELECT
```

```
      pxa=(ADD(pxa,15) MOD 16)+1
      pya=(ADD(pya,15) MOD 16)+1
      pac=TRUE
      IF BTST(lutin%(pyai),16-pxai)
        pac=FALSE
      ENDIF
      pixel_lutin(dx1,dy1,pxai,pyai,noir)
      pac=TRUE
      IF BTST(masque%(pyai),16-pxai)
        pac=FALSE
      ENDIF
      pixel_lutin(dx2,dy2,pxai,pyai,rouge)
      pac=FALSE
    ENDSELECT
  ENDIF
LOOP
END
PROCEDURE resolution
  SELECT XBIOS(4)
  CASE 0
    cx=1
    cy=1
  CASE 1
    cx=2
    cy=1
  CASE 2
    cx=2
    cy=2
  ENDSELECT
RETURN
PROCEDURE init
  LOCAL i
  VSETCOLOR 0,7,7,7
  blanc=0
  VSETCOLOR 1,0,0,0
  noir=1
  VSETCOLOR 2,7,0,0
  rouge=2
  VSETCOLOR 3,0,7,0
  vert=3
  grx=8*cx
  gry=8*cy
  dx1=grx
  dx2=(56+16*8)*cx
  dy1=6
  dy2=6
  corps=4-2*(cy=1)
  nom$="NOMBIDON"
  pac=FALSE
  pxa=1
  pya=1
  mode=1
  coul_masque=blanc
  coul_lutin=noir
  DIM lutin%(16)
  DIM masque%(16)
  DIM chemin$(3)
  DIM extension$(3)
  extension$(1)=".LST"
  extension$(2)=".LST"
  extension$(3)=".LUT"
  FOR i=1 TO 3
    chemin$(i)=CHR$(65+GEMDOS(25))+":"+DIR$(0)+"\*"+extension$(i)
  NEXT i
RETURN
PROCEDURE init_disquette
  LOCAL i,option%
  RESTORE disquette
  disquette$=""
  FOR i=1 TO 37
    READ option%
    disquette$=disquette$+MKI$(option%)
  NEXT i
  disquette:
  DATA 1,1,1,0,1,32764,49158,46874,46874,46874,45082,49146
  DATA 45082,44010,44138,44266,44394,35946,44970,49158,32764
  DATA 0,16376,18660,18660,18660,20452,16388,20452,21524
  DATA 21396,21268,21140,29588,20564,16376,0
RETURN
PROCEDURE init_fleche
  LOCAL i,option%
  RESTORE fleche
  fleche$=""
  FOR i=1 TO 37
    READ option%
    fleche$=fleche$+MKI$(option%)
  NEXT i
  fleche:
  DATA 1,1,1,0,15,49152,48128,17408,17408,17408,31232,1280
  DATA 640,376,136,152,162,231,40,25,10,0,16384,14336,14336
  DATA 14336,1024,512,256,128,112,96,92,24,23,6,4
RETURN
PROCEDURE init_lutin(pxa,pya,mode,coul_masque,coul_lutin,VAR lutin$)
  LOCAL j
  lutin$=MKI$(pxa)+MKI$(pya)+MKI$(mode)
  lutin$=lutin$+MKI$(coul_masque)+MKI$(coul_lutin)
  FOR j=1 TO 16
    lutin$=lutin$+MKI$(0)+MKI$(0)
  NEXT j
RETURN
PROCEDURE cadre_selection
  LOCAL i
  DIM fonction$(10)
  fonction$(1)="F1:INVERSION"
  fonction$(2)="F2:COPIE MASQUE"
  fonction$(3)="F3:SAUVE LUTIN"
  fonction$(4)="F4:SAUVE SOURIS"
  fonction$(5)="F5:SAUVE OBJET"
  fonction$(6)="F6:CHARGE OBJET"
  fonction$(7)="F7:EFFACE OBJET"
  fonction$(8)="F8:"
  fonction$(9)="F9:"
  fonction$(10)="F10:QUITTER"
  DEFTEXT noir,,,4
  DEFFILL blanc
  FOR i=1 TO 5
    PBOX dx1,gry*(16+i),dx1+16*grx,gry*(16+i)+7*cy
    TEXT dx1,gry*(16+i)+5*cy,fonction$(i)
    PBOX dx2,gry*(16+i),dx2+16*grx,gry*(16+i)+7*cy
    TEXT dx2,gry*(16+i)+5*cy,fonction$(i+5)
  NEXT i
RETURN
REM affichage du point d'action
PROCEDURE p_action
```

```
LOCAL bxi,byi,bxf,byf
bxi=dx1+(pxa-1)*grx+1
byi=dy1+(pya-1)*gry+1
bxf=bxi+grx-2
byf=byi+gry-2
DEFFILL noir,2,11
PBOX bxi,byi,bxf,byf
IF BTST(masque%(pya),16-pxa)
  DEFFILL rouge,2,24
  PBOX bxi,byi,bxf,byf
ENDIF
IF BTST(lutin%(pya),16-pxa)
  DEFFILL noir,2,5
  PBOX bxi,byi,bxf,byf
ENDIF
bxi=bxi+(dx2-dx1)
bxf=bxf+(dx2-dx1)
DEFFILL noir,2,11
PBOX bxi,byi,bxf,byf
IF BTST(lutin%(pya),16-pxa)
  DEFFILL noir,2,5
  PBOX bxi,byi,bxf,byf
ENDIF
IF BTST(masque%(pya),16-pxa)
  DEFFILL rouge,2,24
  PBOX bxi,byi,bxf,byf
ENDIF
RETURN
PROCEDURE grille(dx,dy)
  LOCAL pxi,pyi,pxf,pyf,i
  FOR i=0 TO 16
    pxi=dx
    pyi=dy+i*gry
    pxf=dx+16*grx
    pyf=pyi
    LINE pxi,pyi,pxf,pyf
    pxi=dx+i*grx
    pyi=dy
    pxf=pxi
    pyf=dy+16*gry
    LINE pxi,pyi,pxf,pyf
  NEXT i
RETURN
PROCEDURE affiche_lutin
  cadre_lutin(dx1+16*grx+2*cx,dy1,blanc)
  SPRITE lutin1$,dx1+(pxa-1)+16*grx+5*cx,dy1+(pya-1)+3*cy
  cadre_lutin(dx2-22*cx,dy2,noir)
  SPRITE lutin2$,dx2+(pxa-1)-18*cx,dy2+(pya-1)+4*cy
RETURN
PROCEDURE cadre_lutin(dx,dy,coul)
  DEFFILL coul,2,8
  PBOX dx,dy,dx+20*cx,dy+20*cy
  BOX dx,dy,dx+20*cx,dy+20*cy
RETURN
PROCEDURE calcul_lutin
  LOCAL j
  lutin1$=""
  lutin1$=MKI$(pxa)+MKI$(pya)+MKI$(0)
  lutin1$=lutin1$+MKI$(coul_masque)+MKI$(coul_lutin)
  FOR j=1 TO 16
    lutin1$=lutin1$+MKI$(masque%(j))+MKI$(lutin%(j))
  NEXT j
  lutin2$=lutin1$
RETURN
PROCEDURE effacer
  DEFFILL blanc,2,8
  PBOX dx1,dy1,dx1+16*grx,dy1+16*gry
  PBOX dx2,dy2,dx2+16*grx,dy2+16*gry
  grille(dx1,dy1)
  grille(dx2,dy2)
  cadre_lutin(dx1+16*grx+2*cx,dy1,blanc)
  cadre_lutin(dx2-22*cx,dy2,noir)
  p_action
RETURN
PROCEDURE forme_souris(numero)
  SELECT numero
  CASE 8
    DEFMOUSE disquette$
  CASE 9
    DEFMOUSE fleche$
  DEFAULT
    DEFMOUSE (numero)
  ENDSELECT
RETURN
PROCEDURE pixel_lutin(dx,dy,i,j,coul)
  LOCAL bxi,byi,bxf,byf,couleur,motif
  IF (i>0 AND i<17) AND (j>0 AND j<17)
    bxi=dx+(i-1)*grx+1
    byi=dy+(j-1)*gry+1
    bxf=bxi+grx-2
    byf=byi+gry-2
    IF es=2 OR pac=TRUE
      couleur=blanc
      motif=8
    ELSE
      couleur=coul
      motif=8+7*(coul=2)
    ENDIF
    SELECT coul
    CASE 1
      IF couleur=blanc
        lutin%(j)=BCLR(lutin%(j),16-i)
      ELSE
        lutin%(j)=BSET(lutin%(j),16-i)
      ENDIF
      IF (NOT BTST(lutin%(j),16-i)) AND BTST(masque%(j),16-i)
        couleur=2
        motif=1
      ENDIF
      DEFFILL couleur,2,motif
      PBOX bxi,byi,bxf,byf
      IF (NOT BTST(masque%(j),16-i))
        bxi=bxi+(dx2-dx1)*((coul=2)-(coul=1))
        bxf=bxf+(dx2-dx1)*((coul=2)-(coul=1))
        PBOX bxi,byi,bxf,byf
      ENDIF
    CASE 2
      IF couleur=blanc
        masque%(j)=BCLR(masque%(j),16-i)
      ELSE
        masque%(j)=BSET(masque%(j),16-i)
      ENDIF
      IF (NOT BTST(masque%(j),16-i)) AND BTST(lutin%(j),16-i)
        couleur=1
```

```
      ENDIF
      DEFFILL couleur,2,motif
      PBOX bxi,byi,bxf,byf
      IF (NOT BTST(lutin%(j),16-i))
        bxi=bxi+(dx2-dx1)*((coul=2)-(coul=1))
        bxf=bxf+(dx2-dx1)*((coul=2)-(coul=1))
        PBOX bxi,byi,bxf,byf
      ENDIF
    ENDSELECT
    calcul_lutin
    affiche_lutin
    p_action
  ENDIF
RETURN
REM version réduite de affiche_grille
PROCEDURE choix_grille(dx,dy,x,y,coul)
  LOCAL bxi,byi,bxf,byf,i,j
  i=INT((x-dx)/grx)+1
  j=INT((y-dy)/gry)+1
  pixel_lutin(dx,dy,i,j,coul)
RETURN
PROCEDURE decoupe(fichier$,ext_def$,VAR nom$,ext$)
  LOCAL apos|
  apos|=INSTR(fichier$,".")
  IF apos|=0
    ext$=ext_def$
    nom$=fichier$
  ELSE
    ext$=RIGHT$(fichier$,LEN(fichier$)-apos|+1)
    nom$=LEFT$(fichier$,apos|-1)
  ENDIF
RETURN
PROCEDURE analyse_chemin(chem$,ext_def$,VAR lect$,rept$,nom$,ext$)
  LOCAL apos1|,apos2|
  chem$=UPPER$(chem$)
  IF chem$<>""
    apos1|=INSTR(chem$,"\")
    apos2|=RINSTR(chem$,"\")
    lect$=LEFT$(chem$,apos1|-1)
    rept$=MID$(chem$,apos1|,apos2|-apos1|+1)
    decoupe(RIGHT$(chem$,LEN(chem$)-apos2|),ext_def$,nom$,ext$)
    SELECT LEN(lect$)
    CASE 2
      SELECT lect$
      CASE "A:" TO "P:"
      DEFAULT
        lect$=CHR$(65+GEMDOS(25))+":"
      ENDSELECT
    DEFAULT
      lect$=CHR$(65+GEMDOS(25))+":"
    ENDSELECT
  ELSE
    nom$=""
  ENDIF
RETURN
REM sauve lutin, souris, format éditable
PROCEDURE sauve_objet(type)
  LOCAL choix$,ligne$,i,lect$,rept$,ext$
  FILESELECT chemin$(type),nom$+extension$(type),choix$
  analyse_chemin(choix$,extension$(type),lect$,rept$,choix$,ext$)
  IF choix$<>""
    chemin$(type)=lect$+rept$+"*"+ext$
    nom$=choix$
    forme_souris(8)
    OPEN "o",#1,lect$+rept$+nom$+ext$
    SELECT type
    CASE 1,2
      PRINT #1,"PROCEDURE init_"+nom$
      PRINT #1,"LOCAL i&,option%"
      PRINT #1,"RESTORE "+nom$
      PRINT #1,nom$+"$="""
      PRINT #1,"FOR i&=1 TO 37"
      PRINT #1,"READ option%"
      PRINT #1,nom$+"$="+nom$+"$+MKI$(option%)"
      PRINT #1,"NEXT i&"
      PRINT #1,nom$+":"
      ligne$="DATA "+STR$(pxa)+","+STR$(pya)+","+STR$(mode)+","
      ligne$=ligne$+STR$(coul_masque)+","+STR$(coul_lutin)
      PRINT #1,ligne$
      IF type=1
        FOR i=1 TO 16 STEP 2
          ligne$="DATA "+STR$(masque%(i))+","+STR$(lutin%(i))
          ligne$=ligne$+","+STR$(masque%(i+1))+","+STR$(lutin%(i+1))
          PRINT #1,ligne$
        NEXT i
      ELSE
        FOR i=1 TO 16 STEP 4
          ligne$="DATA "+STR$(masque%(i))+","+STR$(masque%(i+1))
          ligne$=ligne$+","+STR$(masque%(i+2))+","+STR$(masque%(i+3))
          PRINT #1,ligne$
        NEXT i
        FOR i=1 TO 16 STEP 4
          ligne$="DATA "+STR$(lutin%(i))+","+STR$(lutin%(i+1))
          ligne$=ligne$+","+STR$(lutin%(i+2))+","+STR$(lutin%(i+3))
          PRINT #1,ligne$
        NEXT i
      ENDIF
      PRINT #1,"RETURN"
    CASE 3
      WRITE #1,pxa,pya,mode,coul_masque,coul_lutin
      FOR i=1 TO 16
        WRITE #1,masque%(i),lutin%(i)
      NEXT i
    ENDSELECT
    CLOSE #1
  ENDIF
  forme_souris(9)
RETURN
REM Pour un nouvel objet
PROCEDURE affiche_grille
  LOCAL bxi,byi,bxf,byf,couleur,motif,i,j
  effacer
  calcul_lutin
  affiche_lutin
  FOR i=1 TO 16
    FOR j=1 TO 16
      couleur=blanc
      motif=8
      bxi=dx1+(i-1)*grx+1
      byi=dy1+(j-1)*gry+1
      bxf=bxi+grx-2
      byf=byi+gry-2
      IF BTST(masque%(j),16-i)
        DEFFILL rouge,2,1
```

```
      PBOX bxi,byi,bxf,byf
    ENDIF
    IF BTST(lutin%(j),16-i)
      DEFFILL noir,2,8
      PBOX bxi,byi,bxf,byf
    ENDIF
    bxi=bxi+(dx2-dx1)
    bxf=bxf+(dx2-dx1)
    IF BTST(lutin%(j),16-i)
      DEFFILL noir,2,8
      PBOX bxi,byi,bxf,byf
    ENDIF
    IF BTST(masque%(j),16-i)
      DEFFILL rouge,2,1
      PBOX bxi,byi,bxf,byf
    ENDIF
  NEXT j
 NEXT i
 p_action
RETURN
PROCEDURE charge_objet
 LOCAL choix$,i,lect$,rept$,ext$
 FILESELECT chemin$(3),nom$+extension$(3),choix$
 analyse_chemin(choix$,extension$(3),lect$,rept$,choix$,ext$)
 IF choix$<>""
   chemin$(3)=lect$+rept$+"*"+ext$
   forme_souris(8)
   nom$=choix$
   OPEN "i",#1,lect$+rept$+nom$+ext$
   INPUT #1,pxa,pya,mode,coul_masque,coul_lutin
   FOR i=1 TO 16
     INPUT #1,masque%(i),lutin%(i)
   NEXT i
   affiche_grille
   CLOSE #1
 ENDIF
 forme_souris(9)
RETURN
PROCEDURE efface_objet
 pxa=1
 pya=1
 mode=0
 coul_masque=0
 coul_lutin=15
 ARRAYFILL lutin%(),0
 ARRAYFILL masque%(),0
 effacer
RETURN
PROCEDURE inversion
 LOCAL i,j
 FOR j=1 TO 16
   lutin%(j)=NOT (lutin%(j))
 NEXT j
RETURN
PROCEDURE copie_masque
 LOCAL j
 FOR j=1 TO 16
   masque%(j)=lutin%(j)
 NEXT j
RETURN
```

Utilisation de morceaux Quartet en Basic GfA

Bien que des exemples d'utilisation des morceaux Quartet soient fournis sur les disquettes du logiciel, certains utilisateurs n'ont pu en tirer parti dans leur basic Gfa. Voici donc un petit listing commenté...

Largement inspiré des sources Basic Hi-Soft fournis, ce programme utilise sa propre routine Pexec. En effet, nous avons rencontré certains problèmes lors de l'exécution du Exec(3,...) du GfA, qui nous ont poussés à en faire l'appel directement en assembleur.

Sébastien Mougey

```
' Utilisation de morceaux Quartet
' en Basic GfA 3
'
RESERVE 50000                        !Libère de la mémoire
'
adr_morceau%=MALLOC(10000)           !Réserve mémoire morceau
adr_voix%=MALLOC(200000)             !Réserve mémoire voix
'
nom$="SINGSONG.PRG"+CHR$(0)
FOR f%=0 TO 40 STEP 2
  READ r%                            !Lecture datas
  DPOKE adr_morceau%+f%,r%
NEXT f%
FOR f%=1 TO LEN(nom$)
  POKE adr_morceau%+40+f%,ASC(MID$(nom$,f%,1))
NEXT f%
'
~C:adr_morceau%()                    !Pexec 3 (Load and not Go)
adr_prog%=LPEEK(adr_morceau%+34)     !Adresse de singsong
adr_call%=adr_prog%+256              !Adresse de lancement
'
BLOAD "DEMO1.4V",adr_morceau%        !Chargement morceau
BLOAD "VOICE.SET",adr_voix%          !Chargement voix
LPOKE adr_call%+12,adr_morceau%      !Pour singsong
LPOKE adr_call%+16,adr_voix%         !(cf doc Quartet)
'
~C:adr_call%()                       !Lancement singsong
'
~MFREE(adr_prog%)                    !Libère mémoire singsong
~MFREE(adr_morceau%)                 !Libère mémoire morceau
~MFREE(adr_voix%)                    !Libère mémoire voix
'
' Routine Pexec, tirée de la disquette "Quartet 2"
DATA &H487A,&H0034,&H487A,&H0030,&H487A,&H001F
DATA &H3F3C,&H0003,&H3F3C,&H004B,&H4E41,&HDEFC
DATA &H0010,&H41fa,&H0006,&H2080,&H4E75,&H0000
DATA &H0000,&H0000,&H0000
```

MICRO VIDEO

La puissance d'une chaîne
La passion d'un spécialiste
☎ : 56.44.47.70 / 56.79.34.89

LOGICIELS DE JEUX A 99 F POUR ATARI ST/STE AMIGA 500 / IBM ET COMPATIBLE PC

LES NOUVEAUTES SUR ATARI ST

A DES PRIX FABULEUX ET AVEC CHAQUE TITRE 1 CADEAU
1 jeu à choisir dans la liste ST ci-contre

F29 ATARI ST	269 F TTC
ROCK STAR(I WAN)	295 F TTC
ITALY 90	249 F TTC
OPERATION STEALTH	315 F TTC
MAUPITI ISLAND	295 F TTC
MIDWINTER	300 F TTC
SIM CITY	300 F TTC
CONQUEROR	290 F TTC
IVANHOE	249 F TTC
GHOST AND GOBLINS	249 F TTC
ULTIMATE GOLF	265 F TTC
GOLD OF AMERICAS	249 F TTC
JUMPING JACK SON	249 F TTC
TENNIS CUP	290 F TTC
COMPTE CHEQUE GESTION FAMILIALE	250 F TTC

LES NOUVEAUTES SUR AMIGA

A DES PRIX FABULEUX ET AVEC CHAQUE TITRE 1 CADEAU
1 jeu à choisir dans la liste (ci contre) amiga à 99 frs

ITALY 90	249 F TTC
F29	269 F TTC
ROCK STAR	295 F TTC
SIM CITY	300 F TTC
JUMPING JACK SON	249 F TTC
TENNIS CUP	290 F TTC

LES NOUVEAUTES SUR PC A DES PRIX FABULEUX ET AVEC CHAQUE TITRE 1 CADEAU

1 jeu à choisir dans la liste (ci-contre) PC

BREAK STONE	290 F TTC
LOOM	320 F TTC
VOYAGEUR DU TEMPS	320 F TTC
FLIGHT SIMULATOR 4	549 F TTC

ATARI ST/STE

AXEL MAGIC HAMMER ST/STE (FRA)
BARBARIAN (PSY) ST/STE (FRA)
CASTLE WARRIOR ST/STE (FRA)
CHUBY CRISTLE ST (FRA)
COMPILATION ACTION ST VOL II ST (FRA)
COMPILATION GEN D'OR ST (FRA)
DEFLECKTOR ST/STE (FRA)
DOUBLE DETENTE ST/STE (FRA)
EXPLORA 1 ST/STE (FRA)
FORGOTTEN WORLD ST/STE (FRA)
GAME SUMMER EDITION ST/STE (FRA)
HATEST (FRA)
HUMAN KILLING MACHINE ST/STE (FRA)
GENIUS ST/STE (FRA)
GARIELD ST (FRA)
INCORRUPTIBLES ST/STE (FRA)
INTERNATIONAL FOOTBALL ST/STE (FRA)
JAWX ST (FRA)
LED STORM ST (FRA)
MANIAX ST/STE (FRA)
MOON WALKER ST/STE (FRA)
MOTOR MASSACRE ST (FRA)
NECRON ST/STE (FRA)
NEVER MIND ST/STE (FRA)
ONSLAUGHT ST/STE (FRA)
OPERATION THUNDER BOLD ST/STE (FRA)
OPERATION WOLF ST (FRA)
OIL IMPERIUM ST/STE (FRA)
RUN THE GAUNTLET ST (FRA)
STORM LORD ST/STE (FRA)
SPACE QUEST 2 ST (ANG)
VICTORY ROAD ST (FRA)
VOYAGEUR ST (FRA)
WIZARD WARZ ST/STE (ANG)
WAR IN THE MEADDLE EARTH ST/STE (FRA)

QUANTITE LIMITEE CHOISISSEZ AU MOINS 2 TITRES DE REMPLACEMENT

AMIGA COMMODORE

ASTAROTH (FRA)
AXEL MAGIC HAMMER (FRA)
BARBARIAN 2 (FRA)
BATMAN (FRA)
CASTLE WARRIOR (FRA)
DATA STORM (ANG)
EXPLORA 1(FRA)
DOUBLE DETENTE (FRA)
FORGOTTEN WORLD (FRA)
INCORRUPTIBLES (FRA)
OIL IMPERIUM (FRA)
ONSLAUGHT (FRA)
OPERATION THUNDERBOLD (FRA)
ROBOCOP (FRA)
STORM LORD (FRA)
SWORD OF TWILIGHT (ANG)

QUANTITE LIMITEE CHOISISSEZ AU MOINS 2 TITRES DE REMPLACEMENT

DISK 5 1/4 PC

CAPONE (ANG)
BILL BUDGE (ANG)
CHARLIE CHAPLIN (ANG)
ALTERNATY REALITY (ANG)
ELITE (ANG)
COMMANDO (ANG) (J)
DRILLER (ANG)
DEEP (FRA)
BAAL (FRA)
DIG DUG (ANG)
NEBULUS (FRA)
GATO (ANG)
DALEY THOMPSON'S (FRA)
DARK SIDE (FRA)
CALIFORNIA GAMES (ANG)
STREET FOOTBALL (ANG)
THE MANHOLE (ANG)
TOP GUN (ANG)
DREAM WARRIOR (FRA)
STREET FIGHTER (FRA)
BIONIC COMMANDO (FRA)
NIGHT RAIDER (ANG)
WIZARD CROW (FRA)
KARTING GRAND PRIX (ANG)
STREET SPORT SOCCER (FRA)
THE GAME SUMMER EDITION(ANG)
RAD WARRIOR (ANG)
SPACE MAX (ANG)
SKY RUNNER (ANG)
UNIVITED (ANG)
COMPILATION(IMPOSSIBLE MISSION,WORLD GAMES,STREET SPORT BASKETBALL) (ANG)
ZOOM (ANG)
WIZARD WARZ (FRA)
LEGACY OF ANCIENTS (ANG)
AAARGH (ANG)
VULCAN (ANG)
ACE (ANG)
WORLD GAMES (ANG)
WAR IN THE MEADDLE EARTH (ANG)
PLATOON (ANG)
PATTON VS ROMMEL (ANG)
SPACE HARRIER (640 KO)
BACK EM (FRA)
FAST BREAKER (FRA)
CHESS MASTER 2000 (FRA)
ACES HIGH COMPILATION (FRA)
MINI PUTT (FRA)
LORD OFF CONQUEST (FRA)
ARTICFFOX (FRA)
WORLD CLASS LEADER BOARD (ANG)
TECHNOCOP (FRA)
THE SHADOW OF MORDER (ANG)
PALADIN (ANG)
ANCIEN LAND OF YS (ANG)
THE TRAIN (FRA)
IMPACT (ANG)
4 TH INCHES (ANG)
CHUCK YEAGERS (FRA)
PALADIN QUEST DISK (ANG)
ORBITER (ANG)
SERVE AND VVOLLEY (FRA)
BARD'S TALE (FRA)
BARD'S TALE II (FRA)
4X4 (ANG)
WORLD TOUR GOLF (FRA)
RACK EM (FRA)
KARATEKA (ANG)
SOLOMON'S KEY (FRA)
HUMAN KILLING MACHINE (FRA)
PEGAGUS (FRA)
TWILIGHT ZONE (ANG)

QUANTITE LIMITEE CHOISISSEZ AU MOINS 2 TITRES DE REMPLACEMENT

NOTE :
(FRA)=DOCUMENTATION FRANCAISE
(ANG)=DOCUMENTATION ANGLAISE

DISK PC 3 1/2 AND 5 1/4

THE BLACK GAULDRON (ANG)
HEAVY BARREL (ANG)
STARRAY(ANG)
SIDE WINDER (ANG)
SILENT SERVICE (ANG)
LOCK ON (ANG)
TIME AND MAGICK (ANG)
SILENT SERVIE (ANG)
ROGGER RABBIT (ANG)
MOTOR BIKE MADNESS (ANG)
FFIRE POWWER (ANG)
ROGGER RABBIT (ANG)
SECRET AGENT(ANG)

HOUSSES

FRAIS D'ENVOI FORFAIT 15 ,00 F

HOUSSE TRANSPORT DISQUETTES 3"1/2 49,00F	HOUSSE PORTFOLIO 159,00F
HOUSSE MEGAST1,520 1040,AMIGA 500 159,00F	HOUSSE CONSOLE LYNX 129,00F

POCHETTE DOMAINE PUBLIC + 30 LOGICIELS 149,00 F POUR ATARI STE/ST ET AMIGA

DISQUETTES VIERGES 3"1/2 DEMARQUEES DF/DD

60F LES 10	280F LES 50
500F LES 100	2450 F LES 500

AVEC BOITES ET ETIQUETTES
FRAIS D'ENVOI FORFAIT 30,00F

A RENVOYER A **MICRO VIDEO** 3 COURS ALSACE ET LORRAINE 33000 BORDEAUX

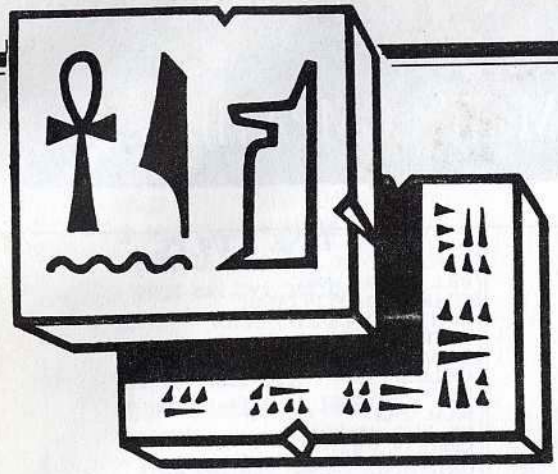
BON DE COMMANDE

FRAIS D'ENVOI ET D'EMBALLAGE
1 A 3 LOGICIELS 25 f
5 A 9 LOGICIELS 50 f
10 LOGICIELS =FRANCO
AJOUTER LES FRAIS D'ENVOI A VOTRE COMMANDE

DESIGNATION	QTE	PX UNIT	PRIX TOTAL
		PORT	
		TOTAL	

Nom.....Prénom.....
adresse.....
code postal.....ville.....
chèque ❒ ou mandat carte ❒

Offres valables dans la limite des stocks disponibles



REALISATION D'UN NOYAU MULTITACHE (SUITE ET FIN)

Il y a deux numéros, lorsque nous avons commencé notre noyau multitâche, il nous a fallu introduire plusieurs notions nouvelles, le programme jouant plutôt un rôle de support. Ce mois-ci, nous allons délaisser un peu la théorie pour nous concentrer un peu plus sur le programme. Nous allons donc implanter des fonctions nouvelles telles que la possibilité de stopper/ redémarrer une tâche ou bien encore la possibilité d'attribuer à chaque tâche une priorité.

LA NOTION DE PRIORITE

Lorsqu'on pense priorité, on pense immédiatement aux interruptions. Une interruption de niveau 2 peut être interrompue par une interruption de niveau 3 qui peut elle-même être interrompue par une interruption de niveau 4, etc. Dans le cas qui nous concerne, ce n'est pas tout à fait la même idée. En effet, il ne s'agit pas de donner la possibilité à une tâche d'en interrompre une autre mais plutôt de donner plus d'importance à une tâche qu'à une autre.

Il existe plusieurs façons de privilégier une tâche. On peut, par exemple, dans le cadre des ressources non-partageables, faire en sorte que si deux tâches demandent l'imprimante, c'est la tâche prioritaire qui l'obtient. Mais cette solution dépasse largement le cadre de notre programme. La méthode que nous allons employer consiste à attribuer plus de temps machine à une tâche prioritaire, et ainsi, une tâche de priorité 2 "tournera" 2 fois plus vite qu'une tâche de priorité 1. Nous allons voir comment la programmer.

GERER LES PRIORITES

La programmation des priorités telle que nous l'avons définie ne pose aucun problème particulier. Souvenez-vous de nos histoires de "Time Slice". Si l'on veut attribuer la priorité 2 à une tâche, il suffit de lui donner la main pendant 2 "tranches de temps" au lieu d'une. Dans notre programme, les étiquettes PRIO et PRIOCOUR ont été rajoutées. PRIO représente 5 mots qui sont la priorité respective de chaque tâche. PRIOCOUR représente la priorité de la tâche courante. Cette valeur est initialisée au début d'une nouvelle tâche et décrémentée à chaque "time slice". Lorsqu'elle arrive à 0, on passe à la tâche suivante. Reportez-vous à l'étiquette NOYAU pour voir les quelques modifications qui ont été nécessaires. Nous allons voir maintenant comment stopper une tâche et la redémarrer.

STOPPER/REDEMARRER UNE TACHE

Implanter de telles fonctions correspond à rendre notre moniteur un peu moins rudimentaire. En effet, la tâche 0 (puisque c'est d'elle qu'il s'agit) se contentait jusqu'à maintenant de stopper le programme lorsqu'on presse la touche ESPACE. Nous allons traiter quatre nouvelles touches A, Z, E et R, qui sont respectivement dédiées aux tâches 1, 2, 3 et 4. La pression d'une de ces touches permet, soit de stopper la tâche si elle est active, soit de la redémarrer si elle est déjà stoppée. Il faut noter que la structure des tâches a été un peu modifiée par rapport à notre précédent article, car il était jusqu'à maintenant impossible qu'une tâche se termine (elle devait reboucler sur elle-même). Ce n'est plus le cas, il suffit que la tâche

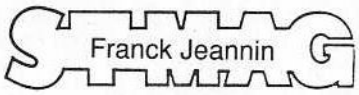
se termine par BRA ENDTACHE et le temps CPU qui lui était alloué sera restitué aux autres tâches. Ainsi, sans que l'utilisateur ne fasse rien, toutes les tâches finiront par être stoppées. Nous allons donc distinguer 3 cas :

* on interrompt une tâche ;
* on redémarre une tâche précédemment stoppée ;
* on redémarre une tâche alors qu'elle était arrivée à son terme.

Pour cela, nous avons utilisé 2 tables différentes, ONOFF qui indique si une tâche est ON ou OFF, et STOP qui nous permet de savoir si une tâche est stoppée ou non. Notre "noyau" se verra donc attribuer la fonction supplémentaire de vérifier que la tâche à laquelle il souhaite donner la main n'est pas stoppée.

La seule fonction qui appelle, peut-être, un commentaire est celle qui redémarre une tâche alors qu'elle était terminée. En fait, c'est très simple, il suffit de procéder comme à l'initialisation de la tâche puis de la déclarer ON et non STOPpée.

Voilà, c'est ainsi que se termine notre exploration du multitâche, en espérant qu'elle aura suscitée en vous la curiosité d'aller plus loin.

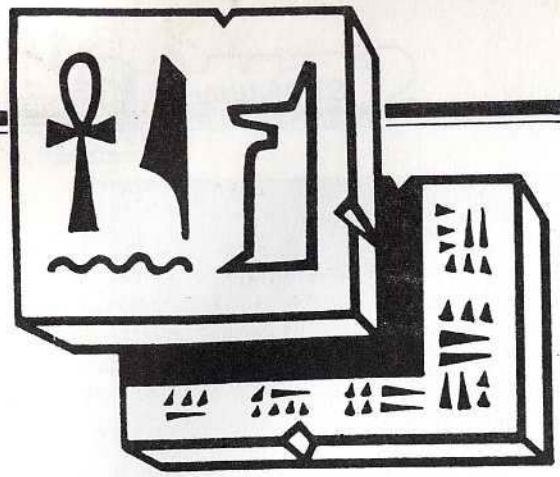



```
; MULTITA2.IS
; ST Mag/Computer's Dream 1990
; Ecrit en GfA Assembleur, fonctionne
; sur écran couleur de préférence
; utiliser les touches A Z E R pour
; stopper et relancer les tâches 1 2 3 4.

.TEXT
  lea.l    pile,sp

  clr.l    -(sp);passer en superviseur
  move.w   #$20,-(sp)
  trap     #1
  addq.l   #6,sp
  move.l   d0,ex_pile

  move.w   #3,-(sp)
  trap     #14        ;xbios(3)
  addq.l   #2,sp
  move.l   d0,sys00  ;adresse physique
  move.l   d0,sys01  ;adresse logique
  ;software end of interrupt
  bclr     #3,$fffa17
  ;on coupe it clavier
  bclr     #6,$fffa09

  bsr      initache

  movea.l  #pilet0,sp
  move.w   #$2700,sr   ;it interdites
  move.l   #noyau,$110 ;timer D
  bset     #4,$fffa15
  bset     #4,$fffa09
  ori.b    #7,$fffa1d
  move.b   #12,$fffa25  ;1 KHz

  move.w   prio,priocour
  move.w   #$2300,sr
  bra      tache0

fin:
;on coupe timer A
  bclr     #4,$fffa09
;autorise it clavier
  bset     #6,$fffa09

quit:
  move.l   ex_pile,-(sp)  ;user
  move.w   #$20,-(sp)
  trap     #1
  addq.l   #6,sp

  clr.w    -(sp)
  trap     #1         ;fin
;
; initache
; initialisation des tâches
;
initache:
  lea.l    onoff,a0
  move.l   #-1,(a0)+
  move.l   #-1,(a0)+
  move.w   #-1,(a0)+

  lea.l    stop,a1
  clr.l    (a1)+
  clr.l    (a1)+
  clr.w    (a1)+

  clr.w    tachecour
  lea.l    ppile,a0
  move.l   #pilet0,(a0)+

  lea.l    pilet1,a1
  move.l   #tache1,-(a1) ;pc
  move.w   #$2300,-(a1)  ;sr
  ;emplacement pour 15 registres
  lea.l    -60(a1),a1
  move.l   a1,(a0)+

  lea.l    pilet2,a1
  move.l   #tache2,-(a1) ;pc
  move.w   #$2300,-(a1)  ;sr
  ;emplacement pour 15 registres
  lea.l    -60(a1),a1
  move.l   a1,(a0)+

  lea.l    pilet3,a1
  move.l   #tache3,-(a1)  ;pc
  move.w   #$2300,-(a1)   ;sr
  ;emplacement pour 15 registres
  lea.l    -60(a1),a1
  move.l   a1,(a0)+

  lea.l    pilet4,a1
  move.l   #tache4,-(a1)   ;pc
  move.w   #$2300,-(a1)    ;sr
  ;emplacement pour 15 registres
  lea.l    -60(a1),a1
  move.l   a1,(a0)+

rts

noyau:
  subq.w   #1,priocour
  bne      rte

  movem.l  d0-d7/a0-a6,-(sp)
  move.w   tachecour,d0
  lsl.w    #2,d0
  lea.l    ppile,a0
  move.l   sp,0(a0,d0.w)

nexttache:
  addq.w   #1,tachecour
  cmpi.w   #4,tachecour
  ble      e000
  clr.w    tachecour
e000:
  move.w   tachecour,d0
  add.w    d0,d0
  lea.l    stop,a0
  tst.w    0(a0,d0.w)
  bne      nexttache

  lea.l    prio,a0
  move.w   0(a0,d0.w),priocour

  add.w    d0,d0
  lea.l    ppile,a0
  movea.l  0(a0,d0.w),sp

  movem.l  (sp)+,d0-d7/a0-a6
rte:
  rte
;
; tache 0
;
tache0:
  moveq.l  #0,d0
  move.b   $fffc02,d0

  cmpi.w   #$39,d0
  beq      fin
```

```
  cmpi.w   #$10,d0
  beq      ctrl1
  cmpi.w   #$11,d0
  beq      ctrl2
  cmpi.w   #$12,d0
  beq      ctrl3
  cmpi.w   #$13,d0
  beq      ctrl4
  bra      tache0

ctrl1:
  tst.w    onoff+2
  beq      relance1
  not.w    stop+2

  bsr      relache
  bra      tache0
relance1:

  lea.l    pilet1,a0

  move.l   #tache1,-(a0)
  move.w   #$2300,-(a0)

  move.w   #-1,onoff+2
  clr.w    stop+2
  bsr      relache
  bra      tache0

ctrl2:
  tst.w    onoff+4
  beq      relance2
  not.w    stop+4

  bsr      relache
  bra      tache0
relance2:

  lea.l    pilet2,a0
  move.l   #tache2,-(a0)
  move.w   #$2300,-(a0)

  move.w   #-1,onoff+4
  clr.w    stop+4
  bsr      relache
  bra      tache0

ctrl3:
  tst.w    onoff+6
  beq      relance3
  not.w    stop+6

  bsr      relache
  bra      tache0
relance3:

  lea.l    pilet3,a0

  move.l   #tache3,-(a0)
  move.w   #$2300,-(a0)

  move.w   #-1,onoff+6
  clr.w    stop+6
  bsr      relache
  bra      tache0

ctrl4:
  tst.w    onoff+8
  beq      relance4
  not.w    stop+8

  bsr      relache
  bra      tache0
relance4:

  lea.l    pilet4,a0

  move.l   #tache4,-(a0)
  move.w   #$2300,-(a0)

  move.w   #-1,onoff+8
  clr.w    stop+8
  bsr      relache
  bra      tache0

relache:
  moveq.l  #0,d0
  move.b   $fffc02,d0
  btst     #7,d0
  beq      relache
  rts
;
; tâche 1
;
tache1:
  moveq.l  #100,d7
b001b:
  movea.l  sys00,a0

  moveq.l  #99,d1
b001:
  moveq.l  #39,d0
b002:
  not.w    (a0)+
  dbra     d0,b002
  lea.l    80(a0),a0
  dbra     d1,b001
  dbra     d7,b001b
  bra      endtache
;
; tâche 2
;
tache2:
  moveq.l  #100,d7
b003c:
  movea.l  sys00,a0
  lea.l    80(a0),a0

  moveq.l  #99,d1
b003:
  moveq.l  #39,d0
b004:
  clr.w    (a0)+
  dbra     d0,b004
  lea.l    80(a0),a0
  dbra     d1,b003

  bsr      pause

  movea.l  sys00,a0
  lea.l    80(a0),a0

  moveq.l  #99,d1
b003b:
  moveq.l  #39,d0
b004b:
  move.w   #-1,(a0)+
  dbra     d0,b004b
  lea.l    80(a0),a0
  dbra     d1,b003b

  bsr      pause
  dbra     d7,b003c
  bra      endtache
;
; tâche 3
;
tache3:
  moveq.l  #15,d2
b005b:
  move.w   #159,d0
b005:
  move.w   #99,d1
b006:

  movem.l  d0-d7/a0-a6,-(sp)
  addi.w   #100,d1
  bsr      point
  movem.l  (sp)+,d0-d7/a0-a6

  dbra     d1,b006
  dbra     d0,b005
  dbra     d2,b005b
  bra      endtache
;
; tâche 4
;
tache4:
  moveq.l  #15,d2
b007b:
  moveq.l  #99,d1
b007:
  move.w   #159,d0
b008:
  movem.l  d0-d7/a0-a6,-(sp)
  addi.w   #160,d0
  addi.w   #100,d1
  bsr      point
  movem.l  (sp)+,d0-d7/a0-a6

  dbra     d0,b008
  dbra     d1,b007
  dbra     d2,b007b
  bra      endtache
;
; point ; affichage d'un point
```

```
; entrée:  d0 position x
;          d1 position y
;          d2 couleur
;
point:
  movea.l  sys00,a0
  mulu.w   #160,d1
  adda.l   d1,a0
  move.w   d0,d1
  andi.w   #$f,d0 ;No du bit ds le mot
  andi.w   #$fff0,d1
  lsr.w    #1,d1
  lea.l    0(a0,d1.w),a0   ;adresse
  neg.w    d0
  addi.w   #15,d0

  move.w   (a0),d1
  btst     #0,d2
  beq      e010
  bset     d0,d1
  bra      e011
e010:
  bclr     d0,d1
e011:
  move.w   d1,(a0)+

  move.w   (a0),d1
  btst     #1,d2
  beq      e012
  bset     d0,d1
  bra      e013
e012:
  bclr     d0,d1
e013:
  move.w   d1,(a0)+

  move.w   (a0),d1
  btst     #2,d2
  beq      e014
  bset     d0,d1
  bra      e015
e014:
  bclr     d0,d1
e015:
  move.w   d1,(a0)+

  move.w   (a0),d1
  btst     #3,d2
  beq      e016
  bset     d0,d1
  bra      e017
e016:
  bclr     d0,d1
e017:
  move.w   d1,(a0)+

  rts
;
; pause
;
pause:
  move.w   #32000,d0
```

```
bpause:
  nop
  nop
  nop
  dbra     d0,bpause
  rts
;
; endtache
; arrête une tâche
;
endtache:
  move.w   tachecour,d7
  add.w    d7,d7
  lea.l    stop,a0
  lea.l    onoff,a1
  clr.w    0(a1,d7.w)      ;off
  move.w   #-1,0(a0,d7.w) ;stoppe

waittimer:
  bra.s    waittimer

  .DATA
;priorité de chaque tâche
prio:
  .DC.w    1,1,2,1,3

  .BSS
ex_pile:
  .DS.l    1

sys00::
  .DS.l    1 ;écran physique
sys01::     .DS.l    1 ;écran logique

tachecour:
  .DS.w    1 ;No de tâche courante
;emplacement pour les 5 pointeurs de pile
ppile:
  .DS.l    5

debutpile:
  .DS.l    500
pile:
  .DS.l    500
pilet0:
  .DS.l    500
pilet1:
  .DS.l    500
pilet2:
  .DS.l    500
pilet3:
  .DS.l    500
pilet4:
;état on ou off de chq tâche
onoff:    .DS.w 5
;état stoppé ou non de chq tâche
stop:     .DS.w 5

;priorité de la tâche courante
priocour: .DS.w 1

  .END
```



LES SCROLLINGS EN GFA

Ce mois-ci, pour clôturer notre série, nous verrons une application des fonctions FGET et FPUT développées le mois dernier, deux routines assembleur pour la réalisation des fonctions miroir vertical et horizontal et une dernière pour le décompactage d'une image DEGAS.

RECTIFICATIF

Une énoooorme erreur a été commise le mois dernier, le codage de la couleur m'ayant échappé. En basse résolution, chaque pixel est codé sur 4 bits appartenant chacun à un plan de couleur (constitué de 1 mot). Une subtilité du système fait que le bit de poids faible du registre se retrouve dans le mot de poids fort et inversement. Pour travailler sur les registres 0 à 8, il faut donc que PLAN%=&x1110 et non pas un bête PLAN%=&x0111. Arghhh!, le remords me ronge.

VOYAGE COSMIQUE

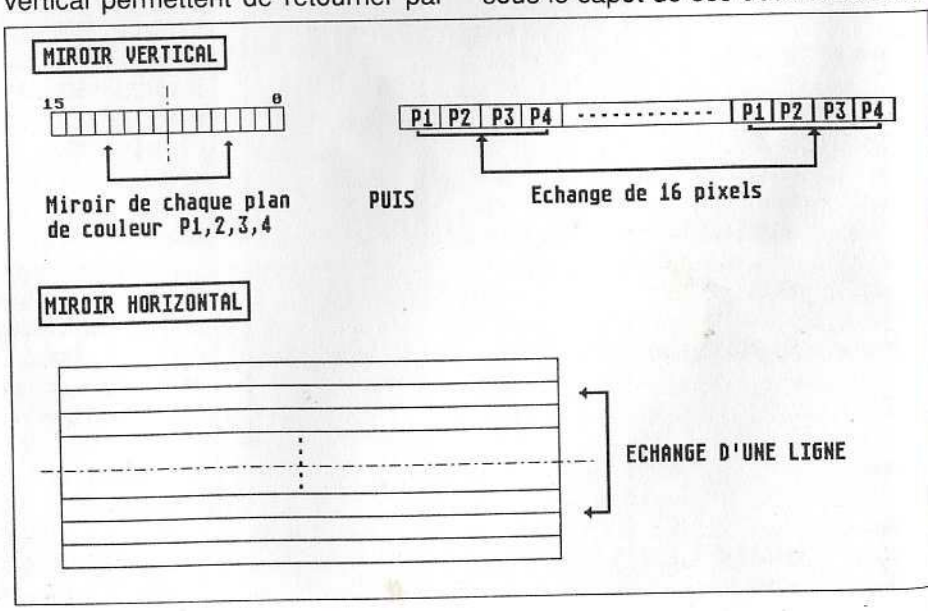
C'est un grand classique des démos. A bord de votre faucon millenium, vous fendez l'espace et voyez les planètes et soleils surgir du fin fond de votre écran. Le même principe est utilisé pour la traversée d'un champ d'astéroïde (comment ça, je délire !). Les astéroïdes grossissent en arrivant sur vous pour atteindre la taille de 64 pixels de côté (sur le pare-brise du vaisseau, ça fait mal). Les différentes tailles de sprites ont été obtenues en rétrécissant le plus gros (64/64) grâce à l'option stretch de DEGAS. Seuls 2 plans de couleur ont été utilisés (Plan%=&x1100, ce qui permet d'utiliser les registres 0,1,2,3 - 0 étant le registre de gauche dans Degas). Cela permet d'être suffisamment rapide pour animer 80 sprites simultanément (de 8 à 80 par les touches de fonction). Vous pouvez tester le programme grâce à la routine DESSINE_ASTEROIDE mais le résultat est beaucoup moins convaincant qu'avec des sprites dessinés à la main. Vous trouverez le listing basic dans le premier encadré intitulé judicieusement ASTEROID.

MIROIR, MIROIR, QUI EST LA PLUS BELLE?

Ces fonctions de miroir horizontal et vertical permettent de retourner par rapport à un axe vertical ou horizontal, un bloc de largeur et de hauteur quelconque (en basse résolution uniquement). L'appel de ces fonction est réalisée par :

```
VOID C:Miror_v%(L:S%,L:H%,L:L%)
```

où S% est l'adresse de l'écran où figure le bloc, H% est la hauteur du bloc en nombre de lignes (1 à 200), et L% la largeur en nombre de paquets de 16 pixels (1 à 20). Ces fonctions sont utiles lorsque les sprites à animer se dirigent dans les deux directions (droite gauche, haut bas). Il suffit alors de dessiner les sprites dans l'une des deux directions et d'en prendre le miroir par la suite. De cette façon, on libère un peu de place sur les disquettes toujours trop chargées. Consultez le listing MIROIR pour l'installation de ces routines. Et comme à l'habitude, ST MAG vous montre ce qu'il y a sous le capot de ces deux fonctions.



Reportez-vous au listing assembleur MIROIR_V et MIROIR_H pour en savoir d'avantage.
Le principe est assez simple. Pour le miroir horizontal, il suffit d'échanger une ligne du haut contre une ligne du bas (avec passage dans un tampon). Pour le miroir vertical, c'est plus délicat. Il faut échanger un paquet de 16 pixels de la gauche par un paquet de la droite en ayant auparavant inversé les bits de chaque plan de couleur (bit 0 à la place du bit 15, etc.).

DECOMPACTAGE DEGAS PI1

C'est promis, je ne vous rappellerai pas le format d'une image DEGAS compactée (reportez vous au numéros 17 page 44 et 23 page 133). Le listing assembleur est la traduction de la routine basic donnée dans le numéro 23. Il suffit de 0,6s à la routine assembleur contre 5 environ pour la version GFA 3 compilée et 40s ! pour la version interprétée. Le décompactage est réalisé par :

```
VOID C:d_degas%(L:image_comp%,
L:image_decomp%).
```

Consultez les listings basic (encadré DEC_DEGA) et assembleur (DECOMP) ainsi que le schéma et rien n'aura plus de secret pour vous (enfin presque). Allez, bonnes vacances !

STupfred

Astéroïd

```
$C+              !Sauve les registres >A3 pour la compilation
HIDEM
@sauve_palette
ON ERROR GOSUB fin
ON BREAK GOSUB fin
@ecran           !on met en place les différents écrans
@load            !installe la fonction de scroll
' @charge_degas       !charge les sprites ou
@dessine_asteroide    !dessine les sprites
@variables
VOID XBIOS(5,L:e2%,L:e2%,W:-1)  !affichage de E2
nb_sprite%=40
DO
  EXIT IF MOUSEK OR INKEY$=" "
  BMOVE e1%,e2%,32000  !transfert de E1 vers E2
  @affiche
  @clavier
  VOID XBIOS(5,L:e3%,L:e2%,W:-1)  !affichage de E2
  VSYNC
  SWAP e2%,e3%    !bascule des 2 écrans physiques
LOOP
@fin
PROCEDURE ecran
  RESERVE FRE(0)-100000
  e1%=GEMDOS(72,L:32000)          !écran qui scrolle
  e21%=GEMDOS(72,L:32260)         !1er écran physique
  e31%=GEMDOS(72,L:32260)         !2em écran physique
  e2%=(e21% AND &HFFFF00)+256
  e3%=(e31% AND &HFFFF00)+256
RETURN
PROCEDURE fin
  fin:
  SHOWM
  VOID XBIOS(6,L:VARPTR(palette$)) !rétablit palette
  VOID XBIOS(5,L:XBIOS(3),L:XBIOS(3),W:-1)
  VOID GEMDOS(73,L:e1%)
  VOID GEMDOS(73,L:e21%)
  VOID GEMDOS(73,L:e31%)          ! on libère la mémoire
  VOID GEMDOS(73,L:fput%)
  VOID GEMDOS(73,L:fget%)
  RESERVE
  END                             !c'est fini
RETURN
PROCEDURE charge_degas
  couleur$=STRING$(32," ")
  OPEN "i",#1,"\asteroid.pi1"
  SEEK #1,2
  BGET #1,VARPTR(couleur$),32   !charge les couleurs
  VOID XBIOS(6,L:VARPTR(couleur$))
  SEEK #1,34
  BGET #1,e1%,32000             !charge l'image
  CLOSE #1
RETURN
PROCEDURE load
  fget%=GEMDOS(72,L:152)   !de la place pour fonction get
  RESTORE fget
  i%=0
  DO
    READ code$
    EXIT IF code$="fin"
    code$="&H"+code$
    code%=VAL(code$)
    fget$=fget$+MKI$(code%)
    INC i%
  LOOP
  BMOVE VARPTR(fget$),fget%,152
  fget:
  DATA 206F,0004,202F,0008,222F,000C,242F,0010,262F,0014
  DATA 246F,0018,282F,001C,49D2,45EA,001E,28C4,E888,E788
  DATA C2FC,00A0,D280,D1C1,3E02,E78F,2C07,4487,0687,0000
  DATA 00A0,5187,28C7,5087,5382,5383,28C2,28C3,70FF,7C03
  DATA 0D04,6700,000E,5280,2206,E389,4481,5C81,38C1,51CE
  DATA FFEC,2C00,5286,E38E,99C6,D9FC,0000,0008,2880,3203
  DATA 2002,7C03,0D04,6700,0004,34D0,41E8,0002,51CE,FFF2
  DATA 51C8,FFEC,D1C7,51C9,FFE4,4E75,fin
  fput%=GEMDOS(72,L:936)   !de la place pour fonction put
  RESTORE fput
  i%=0
  DO
    READ code$
    EXIT IF code$="fin"
    code$="&H"+code$
    code%=VAL(code$)
    fput$=fput$+MKI$(code%)
    INC i%
  LOOP
  BMOVE VARPTR(fput$),fput%,936
  fput:
  DATA 202F,0004,222F,0008,206F,000C,286F,0010,0C40,0140
  DATA 6500,0004,4E75,4A40,6C00,0004,4E75,0C41,00C8,6500
  DATA 0004,4E75,4A41,6C00,0004,4E75,45E8,001E,23E8,0004
  DATA 0005,C64A,23E8,0008,0005,C652,23FC,0000,0000,0005
```

```
  DATA C64E,2C00,0286,0000,000F,4486,0686,0000,0010,13FC
  DATA 0000,0005,C644,E888,2800,2A00,DAA8,0008,0C05,0013
  DATA 6500,0036,4484,0684,0000,0013,23C4,0005,C652,4484
  DATA D8A8,0008,2A28,0018,5285,E38D,CAC4,23C5,0005,C64E
  DATA E78C,D9B9,0005,C64A,13FC,0001,0005,C644,E788,2E28
  DATA 000C,2801,2A01,DAA8,000C,0C45,00C8,6500,000C,4484
  DATA 0684,0000,00C8,2E04,C2FC,00A0,D280,D9C1,2828,0018
  DATA 0C04,0002,6700,00BC,0C04,0001,6700,0184,4A04,6700
  DATA 0226,23C7,0005,C646,2039,0005,C652,7200,7400,7600
  DATA 7800,4241,4242,4243,4244,7A00,4247,3A1A,EDAD,8E85
  DATA 8285,4841,7A00,3A1A,EDAD,8E85,8485,4842,7A00,3A1A
  DATA EDAD,8E85,8685,4843,7A00,3A1A,EDAD,8E85,8885,4844
  DATA 7A00,4847,4647,CF54,835C,CF54,855C,CF54,875C,CF54
  DATA 895C,51C8,FFAE,D5F9,0005,C64E,4A39,0005,C644,6600
  DATA 0022,4847,4841,4842,4843,4844,4647,CF54,835C,CF54
  DATA 855C,CF54,875C,CF54,895C,49EC,FFF8,508C,D9F9,0005
  DATA C64A,2E39,0005,C646,51CF,FF58,4E75,43D4,4BD4,4DD4
  DATA 7800,3828,0010,D9C4,3828,0012,DBC4,3828,0014,DDC4
  DATA 23C7,0005,C646,2039,0005,C652,7200,7400,7600,4241
  DATA 4242,4243,7A00,4247,3A1A,EDAD,8E85,8285,4841,7A00
  DATA 3A1A,EDAD,8E85,8485,4842,7A00,3A1A,EDAD,8E85,8685
  DATA 4843,7A00,4847,4647,CF59,CF59,CF59,CF59,8354,8555
  DATA 8756,508C,508D,508E,51C8,FFB8,D5F9,0005,C64E,4A39
  DATA 0005,C644,6600,0020,4847,4841,4842,4843,4647,CF51
  DATA CF69,0002,CF69,0004,CF69,0006,8354,8555,8756,5089
  DATA 508C,508D,508E,D3F9,0005,C64A,D9F9,0005,C64A,DBF9
  DATA 0005,C64A,DDF9,0005,C64A,2E39,0005,C646,51CF,FF4E
  DATA 4E75,43D4,4BD4,7800,3828,0010,D9C4,3828,0012,DBC4
  DATA 23C7,0005,C646,2039,0005,C652,7200,7400,4241,4242
  DATA 7A00,4247,3A1A,EDAD,8E85,8285,4841,7A00,3A1A,EDAD
  DATA 8E85,8485,4842,7A00,4847,4647,CF59,CF59,CF59,CF59
  DATA 8354,8555,508C,508D,51C8,FFCA,D5F9,0005,C64E,4A39
  DATA 0005,C644,6600,001C,4847,4841,4842,4647,CF51,CF69
  DATA 0002,CF69,0004,CF69,0006,8354,8555,508C,508D,5089
  DATA D3F9,0005,C64A,D9F9,0005,C64A,DBF9,0005,C64A,2E39
  DATA 0005,C646,51CF,FF6E,4E75,43D4,7800,3828,0010,D9C4
  DATA 23C7,0005,C646,2039,0005,C652,7200,4241,7A00,4247
  DATA 3A1A,EDAD,8E85,8285,4841,7A00,4847,4647,CF59,CF59
  DATA CF59,CF59,8354,508C,51C8,FFDC,D5F9,0005,C64E,4A39
  DATA 0005,C644,6600,0018,4847,4841,4647,CF51,CF69,0002
  DATA CF69,0004,CF69,0006,8354,508C,5089,D9F9,0005,C64A
  DATA D3F9,0005,C64A,2E39,0005,C646,51CF,FF8E,4E75,0000
  DATA 0000,0000,0000,0000,0000,0000,0000,0000,fin
RETURN
PROCEDURE sauve_palette
  palette$=SPACE$(32)
  FOR i%=0 TO 30 STEP 2
    DPOKE VARPTR(palette$)+i%,DPEEK(&HFF8240+i%)
  NEXT i%
RETURN
PROCEDURE variables
  l_horiz%=32 !modifiable
  h_horiz%=20 !idem
  horiz_x%=(320-l_horiz%)/2
  horiz_y%=(200-h_horiz%)/2
  nb_sprite%=80 !nb de sprites au maximum
  taillemax%=12 !12 tailles différentes
  DIM x_b%(nb_sprite%),y_b%(nb_sprite%)
  DIM d_x%(nb_sprite%),d_y%(nb_sprite%)
  DIM taille%(nb_sprite%),suc&(nb_sprite%)
  DIM bloc$(taillemax%),bloc%(taillemax%)
  DIM haut%(taillemax%)
  d4%=&X1100    !selection des plans de couleur
  RESTORE asteroide
  FOR i%=taillemax% DOWNTO 1
    READ x%,y%,haut%(i%),larg%        !lit données du sprite
    bloc$(i%)=SPACE$(haut%(i%)*(larg%*8)+30)
    bloc%(i%)=VARPTR(bloc$(i%))       !adresse des sprites
    VOID C:fget%(L:el%,L:x%,L:y%,L:larg%,L:haut%(i%),
              L:bloc%(i%),L:d4%)
  NEXT i%
  asteroide:   !plus gros au plus petit, X,Y,hauteur,largeur
  DATA 0,0,64,4,64,0,56,4,128,0,48,3,176,0,40,3,224,0,32,2
  DATA 256,0,24,2,288,0,16,1,304,0,8,1,0,64,6,1,16,64,4,1
  DATA 32,64,2,1,48,64,1,1
  FOR i%=1 TO nb_sprite%
    taille%(i%)=1
    x_b%(i%)=RANDOM(l_horiz%)+horiz_x%
    y_b%(i%)=RANDOM(h_horiz%)+horiz_y%
    d_x%(i%)=x_b%(i%)-159 !distance en x entre le sprite
                          et le centre de l'ecran
    d_y%(i%)=y_b%(i%)-99  !distance en y
  NEXT i%
  VOID XBIOS(5,L:el%,L:e2%,W:-1)!affichage de E2
  CLS
RETURN
PROCEDURE affiche
  FOR p%=1 TO 12    !affiche les grandes tailles à la fin
    FOR i%=1 TO nb_sprite%
      IF taille%(i%)=p%          IF d_x%(i%)<0
          SUB x_b%(i%),haut%(taille%(i%))/2 !donne
        ELSE                                ! les
          ADD x_b%(i%),haut%(taille%(i%))/2 !coordonnées
        ENDIF                               !du centre
        IF d_y%(i%)<0                       !du sprite
          SUB y_b%(i%),haut%(taille%(i%))/2
        ELSE
          ADD y_b%(i%),haut%(taille%(i%))/2
        ENDIF
        IF x_b%(i%)<0 OR x_b%(i%)>=319 OR y_b%(i%)<0 OR
                  y_b%>=199 OR taille%(i%)=taillemax%
          x_b%(i%)=RANDOM(l_horiz%)+horiz_x%
          y_b%(i%)=RANDOM(h_horiz%)+horiz_y%
          d_x%(i%)=(x_b%(i%)-159)/4
          d_y%(i%)=(y_b%(i%)-99)/4
          taille%(i%)=1             !taille du sprite
        ENDIF
        VOID C:fput%(L:x_b%(i%),L:y_b%(i%),
                  L:bloc%(taille%(i%)),L:e2%)
        ' ADD d_x%(i%),d_x%(i%)/4 !facultatif
        ' ADD d_y%(i%),d_y%(i%)/4 !idem
        ADD x_b%(i%),d_x%(i%)
        ADD y_b%(i%),d_y%(i%)
        INC suc&(i%)
        IF suc&(i%)>=2    !nb de fois avec la même taille
          suc&(i%)=0
          INC taille%(i%)
        ENDIF
      ENDIF
    NEXT i%
  NEXT p%
RETURN
PROCEDURE clavier
  clavier$=INKEY$
```

```
  IF LEN(clavier$)<>1
    clavier%=ASC(RIGHT$(clavier$))
  ELSE
    clavier%=ASC(clavier$)
  ENDIF
  SELECT clavier%
  CASE 59 TO 69        !touches reconnues F1 à F10
    nb_sprite%=(clavier%-58)*8  !de 8 à 80 sprites
  ENDSELECT
RETURN
PROCEDURE dessine_asteroide
  VOID XBIOS(5,L:e1%,L:e2%,W:-1)!dessine sur E1
  SETCOLOR 0,0,0,0
  SETCOLOR 3,5,5,5
  SETCOLOR 2,4,4,4
  SETCOLOR 1,3,3,3
  RESTORE asteroide
  FOR i%=1 TO 12
    READ x%,y%,haut%,larg%       !lit les données
    FOR j%=0 TO haut%/2          !des sprites
      INC couleur%
      IF couleur%=4
        couleur%=1
      ENDIF
      COLOR couleur%
      CIRCLE x%+haut%/2,y%+haut%/2,j%
    NEXT j%
  NEXT i%
  VOID XBIOS(5,L:e2%,L:e2%,W:-1)
RETURN
```

Utilisation des miroirs en GfA

```
On error gosub Fin
On break gosub Fin
@Load        !installe les fonctions MIROIR
@Sauve_palette
@Charge_degas
Rem Miroir vertical et horizontal
S%=Xbios(2)          !adresse source du bloc
H%=200               !hauteur du bloc
L%=20                !largeur du bloc
Do
  Pause 50
  Void C:Miror_h%(L:S%,L:H%,L:L%)
  Pause 50
  Void C:Miror_v%(L:S%,L:H%,L:L%)
  Exit if Mousek Or Inkey$<>""
Loop
@Fin
End
Procedure Load
  Reserve Fre(0)-5000
  Rem Miroir horizontal
  Miror_h%=Gemdos(72,L:184)
  Restore Miror_h
  I%=0
  Do
    Read Code$
    Exit if Code$="fin"
    Code$="&H"+Code$
    Code%=Val(Code$)
    Miror_h$=Miror_h$+Mki$(Code%)
    Inc I%
  Loop
  Bmove Varptr(Miror_h$),Miror_h%,184
  Miror_h:
  Data 206F,0004,202F,0008,222F,000C,43D0,2401,5342,E74A
  Data D3C2,47D0,49D1,5380,2410,6100,0060,2203,2428,0004
  Data 6100,0056,2803,2411,6100,004E,2083,2429,0004,6100
  Data 0044,2143,0004,2281,2344,0004,41E8,0008,43E9,FFF8
  Data B3C8,6200,FFC8,6600,0016,2410,6100,0022,2083,2428
  Data 0004,6100,0018,2143,0004,47EB,00A0,49EC,00A0,41D3
  Data 43D4,51C8,FFA0,4E75,7C00,7E0F,0D02,6600,0008,0F83
  Data 6000,0004,0FC3,5286,51CF,FFEE,7C10,7E1F,0D02,6600
  Data 0008,0F83,6000,0004,0FC3,5286,5387,0C47,000F,6600
  Data FFE8,4E75,fin
  Rem Miroir vertical
  Miror_v%=Gemdos(72,L:72)
  Restore Miror_v
  I%=0
  Do
    Read Code$
    Exit if Code$="fin"
    Code$="&H"+Code$
    Code%=Val(Code$)
    Miror_v$=Miror_v$+Mki$(Code%)
    Inc I%
  Loop
  Bmove Varptr(Miror_v$),Miror_v%,72
  Miror_v:
  Data 206F,0004,202F,0008,222F,000C,43D0,45D0,5380,C0FC
  Data 00A0,D3C0,47D1,5381,2601,2203,2A10,2C28,0004,20D1
  Data 20E9,0004,22C5,22C6,51C9,FFEE,45EA,00A0,47EB,FF60
  Data 41D2,43D3,B3C8,6200,FFDA,4E75,fin
Return
Procedure Fin
  Void Gemdos(73,L:Miror_h%)
  Void Gemdos(73,L:Miror_v%)
  Void Xbios(6,L:Varptr(Palette$)) !rétabli palette
  End
Return
Procedure Charge_degas
  Couleur$=String$(32," ")
  Open "i",#1,"a:\image.pi1"
  Seek #1,2
  Bget #1,Varptr(Couleur$),32   !charge les couleurs
  Void Xbios(6,L:Varptr(Couleur$))
  Seek #1,34
  Bget #1,Xbios(2),32000        !charge l'image
  Close #1
Return
Procedure Sauve_palette
  Palette$=Space$(32)
  For I%=0 To 30 Step 2
    Dpoke Varptr(Palette$)+I%,Dpeek(&HFF8240+I%)
  Next I%
Return
```

Décompression Degas, installation

```
ON BREAK GOSUB fin
ON ERROR GOSUB fin
RESERVE FRE(0)-50000
```

```
@sauve_palette
@load   !installe la fonction de décompactage DEGAS PI1
@charge_degas
INPUT attente$
@fin
PROCEDURE charge_degas
  couleur$=STRING$(32," ")
  FILESELECT "*.pc1","",nom$
  OPEN "i",#1,nom$
  longeur%=LOF(#1)
  image_comp$=SPACE$(LOF(#1))
  image_comp%=VARPTR(image_comp$) !image compactée
  image_decomp%=XBIOS(3) !adresse de l'image décompactée
  SEEK #1,2
  BGET #1,VARPTR(couleur$),32   !charge les couleurs
  VOID XBIOS(6,L:VARPTR(couleur$))
  SEEK #1,34
  BGET #1,VARPTR(image_comp$),longeur%-34 !charge image
  CLOSE #1
  t=TIMER
  VOID C:d_degas%(L:image_comp%,L:image_decomp%)
  PRINT (TIMER-t)/200
RETURN
PROCEDURE fin
  SHOWM
  VOID XBIOS(6,L:VARPTR(palette$)) !rétablit palette
  VOID XBIOS(5,L:XBIOS(3),L:XBIOS(3),W:-1)
  VOID GEMDOS(73,L:d_degas%)
  RESERVE
  END                          !c'est fini
RETURN
PROCEDURE load
  d_degas%=GEMDOS(72,L:182)        !bien à l'abri
  RESTORE d_degas
  i%=0
  DO
    READ code$
    EXIT IF code$="fin"
    code$="&H"+code$
    code%=VAL(code$)
    d_degas$=d_degas$+MKI$(code%)
    INC i%
  LOOP
  BMOVE VARPTR(d_degas$),d_degas%,182
  d_degas:
  DATA 206F,0004,226F,0008,7200,7600,7800,2449,D5FC,0000
  DATA 7D00,7200,1218,0C01,0080,654E,4401,12D0,5203,0C03
  DATA 0028,670E,0C03,0050,6708,0C03,0078,6702,6006,93FC
  DATA 0000,009E,0C03,00A0,6608,93FC,0000,0006,7600,5204
  DATA 0C04,0002,6608,7800,D3FC,0000,0006,51C9,FFC2,D1FC
  DATA 0000,0001,B3CA,65AA,4E75,12D8,5203,0C03,0028,670E
  DATA 0C03,0050,6708,0C03,0078,6702,6006,93FC,0000,009E
  DATA 0C03,00A0,6608,93FC,0000,0006,7600,5204,0C04,0002
  DATA 6608,7800,D3FC,0000,0006,51C9,FFC2,B3CA,6500,FF64
  DATA 4E75,fin
RETURN
PROCEDURE sauve_palette
  palette$=SPACE$(32)
  FOR i%=0 TO 30 STEP 2
    DPOKE VARPTR(palette$)+i%,DPEEK(&HFF8240+i%)
  NEXT i%
RETURN
```

```
PROCEDURE greetings
  DATA Greetings to,Alain le bolivien,l'amirav,Duch,Boul
  DATA The IDN boys,And my little christine
RETURN
```

Miroir vertical

```
;MIROIR VERTICAL BASSE RESOLUTION APPELE DU GFA PAR
;VOID C:ADR%(L:ADR_SOURCE%,L:HAUTEUR%,L:LARGEUR%)
;SANNIER Frédéric, ST MAG 90
      MOVE.L 4(SP),A0  Adresse source du bloc en haut gauche
      MOVE.L 8(SP),D0  Hauteur du bloc
      MOVE.L 12(SP),D1 Largeur du bloc (*16 pixels)
      LEA    (A0),A1   Côté gauche du bloc
      MOVE.L D1,D2
      SUBQ.L #1,D2
      LSL    #3,D2
      ADDA.L D2,A1     Côté droit du bloc
      LEA    (A0),A3
      LEA    (A1),A4
      SUBQ.L #1,D0     -1 pour le test dbra D0
HAUTEUR                Inversion de 16 pixels côté gauche
      MOVE.L (A0),D2   Plan 4 et 3 dans D2
      BSR    MIROIR    Inversion de chaque plan
      MOVE.l D3,D1     On sauve les plans 4 et 3 dans D1
      MOVE.L 4(A0),D2  Plan 2 et 1 dans D2
      BSR    MIROIR    Inversion de chaque plan
      MOVE.L D3,D4     On sauve les plans 2 et 1 dans D4
;inversion de 16 pixels côté droit
      MOVE.L (A1),D2   Plan 4 et 3 dans D2
      BSR    MIROIR    Inversion de chaque plan
      MOVE.l D3,(A0)   Copie des plans 4 et 3 à gauche
      MOVE.L 4(A1),D2  Plan 2 et 1 dans D2
      BSR    MIROIR    Inversion de chaque plan
      MOVE.L D3,4(a0)  Copie des plans 2 et 1 à gauche
      MOVE.l D1,(A1)   Copie des plans 4 et 3 à droite
      MOVE.l D4,4(A1)  Copie des plans 2 et 1 à droite
      LEA    8(A0),A0  16 pixels suivants
      LEA    -8(A1),A1
      CMPA.l A0,A1     Si A1>A0 on continue
      BHI    HAUTEUR
      BNE    LIGNE     Si inférieur on passe à la ligne
      MOVE.L (A0),D2   Cas où A1=A0
      BSR    MIROIR    (Largeur de bloc impaire)
      MOVE.L D3,(A0)   On réalise le miroir des plans
      MOVE.L 4(A0),D2  Mais pas de transfert à réaliser
      BSR    MIROIR    Car il s'agit du centre du bloc
      MOVE.L D3,4(A0)
LIGNE                  Passage à la ligne
      LEA    160(A3),A3
      LEA    160(A4),A4
      LEA    (A3),A0
      LEA    (A4),A1
      DBRA   D0,HAUTEUR
      RTS              Retour au basic
MIROIR                 Sous-programme de miroir des plans
      MOVEQ  #0,D6     Inversion des bits 0 à 15
      MOVEQ  #15,D7
BIT   BTST   D6,D2     Test du bit D6 de D2
      Bne    UN
ZERO  BCLR   D7,D3     Bit égal à 0
      BRA    SU
```

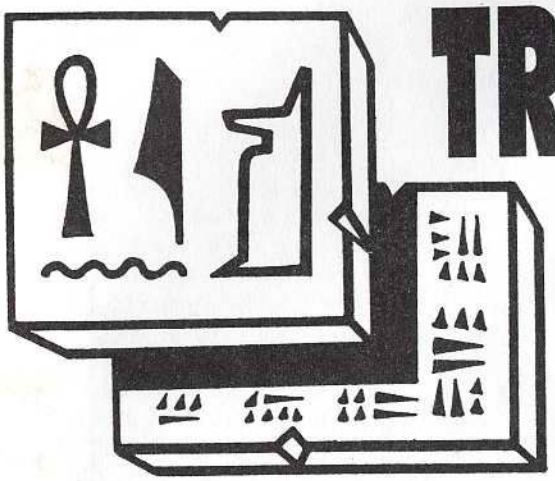
```
UN     BSET   D7,D3      Bit égal à 1
SU     ADDQ.L #1,D6
       DBRA   D7,BIT     Bit suivant
       MOVEQ  #16,D6     Inversion des bits 16 à 31
       MOVEQ  #31,D7
BIT1   BTST   D6,D2
       BNE    UN1
ZERO1  BCLR   D7,D3
       BRA    SU1
UN1    BSET   D7,D3
SU1    ADDQ.L #1,D6
       SUBQ.L #1,D7
       CMPI   #15,D7
       BNE    BIT1
       RTS               Retour du sous programme
```

Miroir horizontal

```
;MIROIR HORIZONTAL BASSE RESOLUTION APPELE DU GFA PAR
;VOID C:ADR%(L:ADR_SOURCE%,L:HAUTEUR%,L:LARGEUR%)
;SANNIER Frédéric, ST MAG 90
       MOVE.L 4(SP),A0      Adresse source du bloc haut gauche
       MOVE.L 8(SP),D0      Hauteur du bloc
       MOVE.L 12(SP),D1     Largeur du bloc (*16 pixels)
       LEA    (A0),A1
       LEA    (A0),A2       Haut gauche du bloc
       SUBQ.L #1,D0
       MULU   #160,D0       Nb de ligne-1 *160 octets
       ADDA.L D0,A1
       LEA    (A1),A3       Bas gauche du bloc
       SUBQ.L #1,D1         -1 pour le test dbra d1
       MOVE.L D1,D3         Largeur du bloc sauvé dans d3
LIGNE
       MOVE.L D3,D1         Rétablit la largeur du bloc
MOT    MOVE.L (A0),D5       Transfert de 16 pixels du haut
       MOVE.L 4(A0),D6      Vers un tampon
       MOVE.L (A1),(A0)+    Transfert de 16 pixels du bas
       MOVE.L 4(A1),(A0)+   Vers le haut
       MOVE.L D5,(A1)+    Transfert de 16 pixels du tampon
       MOVE.L D6,(A1)+      Vers le bas
       DBRA D1,MOT          Pour toute la largeur du bloc
       LEA    160(A2),A2    Passage à la ligne
       LEA    -160(A3),A3
       LEA    (A2),A0
       LEA    (A3),A1
       CMPA.1 A0,A1        Si A1>A0 on permute 2 autres lignes
       BHI    LIGNE         Si A1=A0 axe de symetrie
       RTS                  RETOUR AU BASIC
```

Décompression Degas

```
;DECOMPACTAGE D'IMAGE DEGAS BASSE RESOLUTION APPELE DU GFA
;PAR VOID C:ADR%(L:IM_COMP%,L:IM_DECOMP%)    SANNIER Frédéric
       MOVE.L    4(SP),A0  Adresse image compactée
       MOVE.L    8(SP),A1  Adresse image décompactée
       MOVEQ     #0,D1
       MOVEQ     #0,D3
       MOVEQ     #0,D4
       MOVEA.L   A1,A2
       ADDA.L    #32000,A2
OCT_SUIV
       MOVEQ     #0,D1
       MOVE.B    (A0)+,D1
       CMPI.B    #128,D1
       BLO.S     INF128
       NEG.B     D1
FORG   MOVE.B    (A0),(A1)+
       ADDQ.B    #1,D3
FP=40  CMPI.B    #40,D3
       BEQ.S     EGAL40
       CMPI.B    #80,D3
       BEQ.S     EGAL40
       CMPI.B    #120,D3
       BEQ.S     EGAL40
       BRA.S     PAS40
EGAL40
       SUB.L     #158,A1
PAS40  CMPI.B    #160,D3
       BNE.S     PAS160
       SUBA.L    #6,A1
       MOVEQ     #0,D3
PAS160
       ADDQ.B    #1,D4
       CMPI.B    #2,D4
       BNE.S     PAS2
       MOVEQ     #0,D4
       ADDA.L    #6,A1
PAS2
NEXTG  DBRA D1,FORG
       ADDA.L    #1,A0
       CMPA.1    A2,A1
       BlO.S     OCT_SUIV
       RTS
INF128
FORG1  MOVE.B    (A0)+,(A1)+
       ADDQ.B    #1,D3
FP1=40
       CMPI.B    #40,D3
       BEQ.S     EGAL40b
       CMPI.B    #80,D3
       BEQ.S     EGAL40b
       CMPI.B    #120,D3
       BEQ.S     EGAL40b
       BRA.S     PAS40b
EGAL40b
       SUB.L     #158,A1
PAS40b
       CMPI.B    #160,D3
       BNE.S     PAS160b
       SUBA.L    #6,A1
       MOVEQ     #0,D3
PAS160b
       ADDQ.B    #1,D4
       CMPI.B    #2,D4
       BNE.S     PAS21
       MOVEQ     #0,D4
       ADDA.L    #6,A1
PAS21
NEXTG1
       DBRA D1,FORG1
RETOUR
       CMPA.1    A2,A1
       BlO  OCT_SUIV
       RTS              Retour basic
```



TRUCS ET ASTUCES EN MIDI

L'objet de la série d'articles que nous entamons ici est audacieux : en invitant tous les ataristes à débuter dans la création musicale et la programmation, nous espérons initier les programmeurs récalcitrants aux joies des orifices situés sur le flanc gauche de leur machine tout en diminuant sensiblement le taux de dépressions chez les musiciens de synthèse. Chaque mois, un aspect différent du M.I.D.I. sera traité et mis en pratique dans de petits programmes proposés en basic GfA et Omikron.

PRAELUDIUM

Si les programmeurs confirmés trouveront matière à réflexions sonnantes voire trébuchantes, nous avons visé la simplicité en employant les langages les plus courants. Tous les programmes seront modulaires et dûment commentés ; leur collection fournira à terme une panoplie propre à la création de moult applications musicales.

L'INTERFACE M.I.D.I. : UN PEU D'HISTOIRE

1981 : la marque de synthétiseurs "Sequential Circuit" propose une entente sur la base de son interface, l'U.S.I.
1982 : Dans le double but d'établir un standard de communication entre synthétiseurs et d'affermir le cours du Yen, les japonais s'y intéressent et créent la norme M.I.D.I. (Musical Instrument Digital Interface).

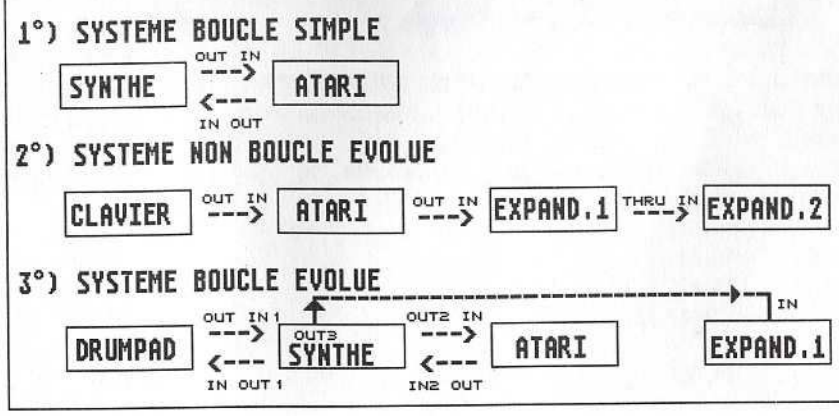
Il y a donc fort à parier, même pour les non anglicistes, que les 2 prises M.I.D.I. 5 broches qui voisinent le port cartouche de votre ST servent à véhiculer de l'information numérique et à faire communiquer les appareils M.I.D.I. entre eux (voir en figure 1, quelques exemples de connexions). Disons pour le moment qu'elles reçoivent (Midi IN) et transmettent (Midi OUT) en série ("à la queue-leu-leu") un train d'informations suivant un protocole précis, à la vitesse honorable de 31,25 Kbauds (soit 31.250 bits par seconde).

Une troisième prise (Midi Thru) se charge de ressortir l'information telle qu'elle entre par la Midi IN, sans traitement ni altération. Le ST n'étant pas pourvu de cette dernière à l'origine, la procédure "THRU" qui accompagne le programme n°1 propose une transformation "soft" (logicielle) de la Midi OUT en Midi THRU (pour la THRU Hardware, consulter ST MAG n°6, p. 49). Une remarque : nous nous sommes servis des fonctions "BIOS" (Basic Output-Input System) prévues pour la gestion des ports MIDI.

LE LANGAGE M.I.D.I. : NOTE ON ET NOTE OFF

Les instruments Midi (synthétiseur, boîte à rythmes, guitare, batterie, saxophone, ordinateur... équipés de cette interface) communiquent entre eux en s'envoyant des informations de plusieurs types. Nous commencerons par les 2 ordres Midi les plus courants : NOTE ON et NOTE OFF (activation et désactivation d'une note).

FIGURE 1 : EXEMPLE DE CONNECTIONS D'APPAREILS M.I.D.I.



Lorsqu'une touche d'un clavier Midi est enfoncée, un train d'information en 3 parties sort de sa prise Midi Out : un premier octet appelé "octet de statut" (StatusByte), suivi de 2 octets de données. Le premier octet nous annonce qu'une touche du clavier vient d'être enfoncée ("Note On", Valeur=144) et nous indique le canal MIDI sur lequel la note est jouée (Valeur : de 0 à 15, ce qui fait bien 16 canaux Midi au total). Ces deux renseignements sont codés sur un seul octet, mais comment ? Ils sont tout simplement additionnés l'un à l'autre ! Pas de panique : pour une note jouée sur le canal 1, le premier octet vaut donc 144+0=144. Sur le canal 2 : 144+1=145, et ainsi de suite jusqu'au canal 16 : 144+15=159.

Le second octet marque la hauteur de la note (Valeur : de 0 à 127), bien que les meilleurs claviers fassent 88 notes quand 95% des synthés se jouent sur 5 octaves ! A titre de repère, le DO 3 (DO de la serrure) vaut 60 et le LA du diapason (LA 440 Hz) vaut 69 ! Le troisième octet indique la vélocité, calculée par la vitesse de frappe sur la touche (Valeur : de 0 à 127).
La figure 2 nous décortique ce "train d'information". Il faut ici clarifier les bases numériques employées : la valeur d'un "NOTE ON" sur le canal MIDI 1 peut se lire (voir la figure 2) en binaire (1001 0000), en décimal (144) ou en hexadécimal (&H90). Les octets de statut (en tête de tous les trains d'information Midi) seront désormais exprimés ici en hexadécimal et précédés du symbole "&H", les valeurs des octets de données (hauteur, vélocité...) restant en décimal.

MIDI A 14 HEURES !

La philosophie du système MIDI repose sur une structure résolument ouverte. Depuis la parution de la "Midi Specification 1.0", la liste des ordres MIDI n'a cessée de s'étoffer, chaque constructeur appuyant ses innovations techniques par l'ajout, notamment, de nouveaux contrôleurs MIDI. Notre première préoccupation sera de vérifier quels ordres votre machine peut comprendre et transmettre : son "Implémentation Midi". La richesse du code Midi nous permet en outre d'exprimer la même chose de plusieurs façons : c'est le cas de l'ordre "NOTE OFF".

Comme pour "NOTE ON", il s'agit d'une information en trois parties. Mais suivant le dialecte du synthé incriminé, le relâchement d'une note sur le canal 1 se dira "&H90 HAUTEUR 0" (une Note On avec Vélocité 0 correspond à une Note Off) ou "&H80 HAUTEUR VELOCITE", étant entendu que le musicien relâche la note selon la VELOCITE de son humeur, de 0 à 127 (voir figure 3). Pire ! Suivant la façon dont un accord sera joué sur le clavier, l'octet de statut "NOTE ON CANAL 1" (&H90) figurera ou non devant la hauteur et la vélocité de chaque note ! (voir figure 3).
Le Midi envoie de surcroît toutes les 0,3 secondes un code pour tester si les connexions restent établies : l'Active Sensing, véritable mouche du coche qu'on visera à museler pour l'heure. On peut s'assurer des particularités de son propre synthé avec le programme n°1, "Etat des Lieux". Mais avant de commencer, on peut réfléchir sur 3 choses : l'implémentation Midi du synthétiseur (codes Midi reconnus par sa machine) ; la configuration de l'appareil (filtrage des codes reçus ou transmis) ; le routage de ses connexions Midi (place de chaque appareil dans la chaîne...).

Pour l'heure, une simple connexion OUT SYNTHE <-> IN ATARI suffit : jouer quelques notes ou manipuler un contrôleur en visualisant les codes produits sur l'écran vaut mieux qu'un long discours. Cette étape est nécessaire pour se faire une idée de la structure des codes MIDI. D'autre part, vous trouverez dans l'encadré ci-après une première liste des principaux codes MIDI que vous risquez de retrouver à l'écran, et qui sont répartis en deux grandes catégories : les Messages Canaux et les Messages Système, cette dernière étant divisée en deux sous-catégories, les Messages Communs et les Messages Exclusifs. Cette liste n'est pour le moment qu'indicative et non commentée, et nous verrons au fur et à mesure de nos mini-développements le détail de toutes les commandes du Midi.

FIGURE 2 : TRAIN D'INFORMATION MIDI

1°) ON APPUIE SUR LA NOTE "DO 3" DE SON SYNTHETISEUR :

1°)	OCTET DE STATUT	OCTET DE DONNEES	OCTET DE DONNEES
	1 0 0 1 0 0 0 0	0 0 1 1 1 1 0 0	0 1 1 1 1 1 1 1
SOIT EN BINAIRE:	1001+0000	00111100	01111111
SOIT EN HEXADEC:	&H90+ 0		
SOIT EN DECIMAL:	144+ 0	DECIMAL=60	DECIMAL=127
SOIT EN CLAIR:	NOTEON,CANAL 1	HAUTEUR=DO3	VELOCITE=127

FIGURE 3 : UN SEUL "NOTE ON" SERVANT A ANNONCER PLUSIEURS NOTES : HAUTEUR ET VELOCITE SUFFISENT A DECRIRE LES NOTES (VELOCITE=0:NOTE OFF) ICI ON APPUYE SUR DO3(60) ET RE3(62) PUIS ON RELACHE LES 2 NOTES .

	Octet de statut	Octet de données	Octet de données
DEBUT DU TRAIN --->	1 0 0 1 0 0 0 0	0 0 1 1 1 1 0 0 (1ère NOTE)	0 1 1 1 1 1 1 1 (1ère NOTE)
	NOTE ON , CANAL 1	HAUTEUR=60	VELOCITE=127

Octet de données	Octet de données	Octet de données	Octet de données
0 0 1 1 1 1 1 0 (2ème NOTE)	0 1 1 1 1 1 1 1 (2ème NOTE)	0 0 1 1 1 1 0 0 (1ère NOTE)	0 0 0 0 0 0 0 0 (1ère NOTE)
HAUTEUR=62	VELOCITE=127	HAUTEUR=60	VELO=0->NOTE OFF

Octet de données	Octet de données	
0 0 1 1 1 1 1 0 (2ème NOTE)	0 0 0 0 0 0 0 0 (2ème NOTE)	<--- FIN DU TRAIN
HAUTEUR=62	VELO=0->NOTE OFF	

1. Les "Channel Messages"

a) Les commandes :

NOTE ON	&H 90 + CANAL, NOTE, VELOCITE
NOTE OFF	&H 80 + CANAL, NOTE, VELOCITE
ou	&H 90 + CANAL, NOTE, 0
POLYPHONIC AFTERTOUCH	&H A0 + CANAL, NOTE, VALEUR
PROGRAM CHANGE	&H C0 + CANAL, N°----SON
CHANNEL AFTERTOUCH	&H D0 + CANAL, VALEUR
PITCH BEND	&H E0 + CANAL, 0, AMPLITUDE

b) Les contrôleurs :

Tous les contrôleurs sont du format :

&H B0 + CANAL, N° CONTROLEUR, VALEUR

Les valeurs respectives de ces trois octets varient entre :

176+(0 à 15), 0 à 127, 0 à 127

Suivent les numéros et les noms des contrôleurs les plus connus :

1. Modulation Wheel	65. Portamento
2. Breath Controller	66. Sustenuto
3. Foot Pedal	67. Soft Pedal
4. Foot Controller	92. Tremolo
5. Portamento Time	93. Chorus
6. Data Entry	94. Celeste
7. Main Volume	95. Phaser
10. Pan	96. Data Increase
11. Expression	97. Data Decrease
64. Hold Sustain	

et les "Messages Mode", inclus dans les Contrôleurs:

122. Local Mode	125. Omni On
123. All Notes Off	126. Poly Off
124. Omni Off	127. Poly On

2. Les "System Messages"

a) Les "Messages Communs" :

&H F2. Song Position	&H FB. Cont
&H F3. Song Selection	&H FC. Stop
&H F6. Tune Request	&H FE. Active Sensing
&H F8. Clock	&H FF. System Reset
&H FA. Start	

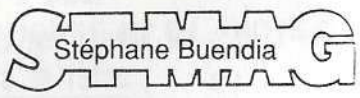
b) Les Exclusifs :

&H F0. SYSTEM EXCLUSIVE
&H F7. SYSTEM END

CONCLUSION

Un coup d'oeil rassure sur la complexité du protocole MIDI : mis à part quelques spécimens tordus qui seront l'objet d'un prochain article (du style du Pitch Bend par exemple), l'important à retenir aujourd'hui est que les messages se présentent sur 3 octets (Code + Canal, Valeur, Valeur) et se répartissent en deux grandes familles : les "Channel messages" et "System messages".

Voilà donc une bonne façon de redécouvrir son synthétiseur : explorer ses fonctions en visualisant ses transmissions. Si vous tombez sur des commandes ou des contrôleurs MIDI non répertoriés sur la liste, notez-les : ils sont propres à votre appareil et vous serviront à construire un éditeur ou à envoyer des messages de configuration (surveiller notamment les Contrôleurs n°96 à 121). Un coup d'oeil sur le manuel du synthé, à la page "Implementation MIDI", ne fera pas de mal non plus. Les choses sérieuses commencent le mois prochain !

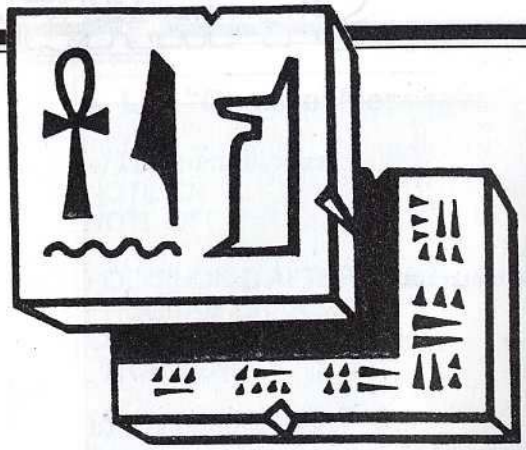


PROGRAMME n° 1,"ETAT des lieux"

```
' Affichage de la Midi In et procedure Midi Thru.
' par Stéphane Buendia, Mai 90.
'
REPEAT
  IF INP?(3)                 !Scrute le port Midi In
    octet=INP(3)             !Attrappe un octet Midi
    OUT 3,octet              !Le sort par le port Midi Out
    octet=octet AND &HFF     !Ne garde que ses 8 bits bas
    IF octet<>&HFE           !Retire l'active sensing
      IF (octet AND &H80)    !Si octet>=&H80,alors statut
        PRINT                !Saute une ligne pour indiquer
      ENDIF                  !l'apparition d'un ordre
      PRINT octet            !Affichage de l'octet
    ENDIF
  ENDIF
UNTIL LEN(INKEY$)            !Ad libitum...
'
END
'
PROCEDURE thru               !La Midi Thru retransmet ce
  REPEAT                     !qui arrive de la Midi In :
    IF INP?(3)               !Si un octet arrive,
      octet=INP(3)           !il est capturé puis
      OUT 3,octet            !immédiatement sorti.
    ENDIF
  UNTIL LEN(INKEY$)
RETURN
```

PROGRAMME N° 2, "Monitoring midi"

```
' Pour vos réceptions Midi guindées.
' Affichage sur invitation uniquement.
' par Stephane Buendia, Mai 90.
'
form$="####"
DIM note$(60)
RESTORE notes
FOR i=1 TO 12
  READ note$(i)
NEXT i
DIM control$(120)
RESTORE controls1
FOR i=1 TO 11
  READ control$(i)
NEXT i
@coup_de_balai
'
drapeau=1
REPEAT
  IF INP?(3)                !Si un octet est présent
    IF drapeau              !Drapeau en position
      @entrez               !Attrape l'octet
    ENDIF
    canal=(x AND &HF)+1     !Canal = quartet bas
    command=x AND &HF0      !Commande = quartet haut
    IF command=&H90         !Si note on
      GOSUB entrez          !On attrape le suivant
      hauteur=x             !(la hauteur de la note)
      '
      REPEAT
        GOSUB entrez        !Le prochain octet = vélocité
        velocite=x
        IF velocite=0       !Note On avec vélocité zéro
          PRINT "OFF",      != Note Off
        ELSE
          PRINT "ON ",
        ENDIF
        @justif(canal)      !Affichage note, canal,
        '                    hauteur et vélocité.
        @affiche_notes
        @justif(velocite)
        PRINT
        GOSUB entrez        !Le "running status" change ?
        hauteur=x           !Oui = nouvel octet de statut
        drapeau=0           !Nous n'attraperons que des
        command=x AND &HF0  !hauteurs et des vélocités.
      UNTIL command>=&H80   !nouvel octet de statut > sort
    ELSE
      drapeau=1
      IF command=&H80       !&H80 : Note Off certifiée.
        GOSUB entrez
        hauteur=x
        GOSUB entrez
        velocite=x
        PRINT "OFF",
        @justif(canal)
        @affiche_notes
        @justif(velocite)
        PRINT
      ENDIF
      IF command=&HB0
        GOSUB entrez
        numero=x
        GOSUB entrez
        valeur=x
        @affichage_des_controleurs
      ENDIF
      '
    ENDIF
  ENDIF
UNTIL LEN(INKEY$)
END
'
PROCEDURE coup_de_balai
  WHILE INP?(3)
    x=INP(3)
  WEND
RETURN
'
PROCEDURE entrez
  REPEAT
    x=INP(3)
  UNTIL x<>&HFE
RETURN
'
PROCEDURE affiche_notes
  PRINT USING "&",note$((hauteur MOD 12)+1);!(hauteur\12)-2
  @justif((hauteur\12)-2)
RETURN
'
PROCEDURE justif(n)         !On comparera sans polémiquer
  PRINT USING "####",n;     !la concision d'un "print
  PRINT ,                   !using" en GfA et en Omikron
RETURN
'
PROCEDURE affichage_des_controleurs
  PRINT "CTR",              !Nous ne proposons ici que la
  @justif(canal)            !liste des 12 premiers
  num$=control$(numero)     !contrôleurs, à vous d'ajouter
  IF numero>0 AND numero<13 !ceux qui vous intéressent
    PRINT USING "&",num$+"                ";
    @justif(valeur)
  ENDIF
  IF numero>12              !Affichage par défaut
    ' @justif(numero)
    PRINT USING "####",numero;
    PRINT "                    ";
    ' PRINT valeur
    @justif(valeur)
  ENDIF
  PRINT
RETURN
'
'     Voici la liste des 11 premiers contrôleurs :
controls1:
DATA MOD_WHEEL,BREATH_CTRL,FOOT_PEDAL,FOOT_CTRL,PORTA_TIME
DATA DATA_ENTRY,MAIN_VOL,8,9,PAN,EXPRESSION
'
notes:                      !respectez les espaces S.V.P.
DATA DO ,DO#,RE ,MIb,MI ,FA ,FA#,SOL,LAb,LA ,SIb, SI
```



LE COIN PROPRE A BILLY T.

Les lois du hasard, la chance, le bol, la baraka... Pour vous et moi, la notion de chance est quelque chose d'assez élastique. Pour les matheux (et donc pour les ordinateurs à la logique inflexible), ces notions ne sont que le voile flou d'une branche rigoureuse et très utile des maths, les probabilités. L'application à la programmation est évidente : le besoin de simuler l'aléatoire est partout en informatique, depuis le simulateur de vol ou de cité aux images fractales, en passant par les jeux, bien sûr. Un peu perdu ? Pas de panique, suivez votre guide, Billy T.

RIONS UN PEU

Vous souvenez-vous de ce débat télévisé qui suivit la dernière rediffusion du film de SF "Météore" ? Un scientifique y affirmait que les chances pour que la terre soit heurtée par une météorite géante étaient de une tous les dix mille ans et que cela faisait bien 30 ou 40 000 ans que cela n'était plus arrivé. Réaction du journaliste : "Alors, les chances augmentent tous les jours ?"... Silence gêné du savant (ou était-ce un fou-rire?).

Autres exemples : des journaux en vente libre affirment qu'il faut jouer le numéro N du loto "car il n'est pas tombé depuis longtemps". Tel homme, père de cinq filles, se réjouit de savoir sa femme de nouveau enceinte, car ce coup-ci, "il y a forcément plus de chances que ce soit un garçon".

Faux. Faux, faux, et faux. Raisonner comme ça, c'est se laisser embobiner par les marchands de martingales ou bien s'exposer à de cruelles désillusions. Certains cultivent cet art du mensonge subtil basé sur l'ignorance des consommateurs et s'en font leurs choux gras. Quand j'en aurais fini avec votre culture probabiliste, mes p'tits gars, vous ne serez plus victimes de ce genre d'escroqueries (Oui, Tontooon!).

ET LA PROGRAMMATION ?

Puisqu'ici, la programmation est le sport national, prenons un exemple concret. Vous êtes en train de déboguer le jeu d'arcade de l'année, musique somptueuse, graphismes raffinés, prix raisonnable. Vous testez donc votre merveille. Une horde de 18 aliens envahit l'affichage 3D, vous allez vous en débarasser, quand soudain, horreur ! Un des monstres est apparu hors de l'écran ! Tout plante ! Bien sûr, vous utilisez la fonction 17 du XBios, qui suffit bien à vos besoin, pensez-vous. Erreur : cette fonction (un générateur de nombres aléatoires, ou GNA) vient de retourner une série de valeurs qui ont entraîné l'activation d'un bug de votre programme. Or, votre GNA ne peut être utilisé pour fournir une deuxième fois la même série de valeurs : il renvoie systématiquement une série différente après chaque allumage. Impossible de l'employer pour un déboguage fiable. Chaque appel à la fonction GNA fournit certes un résultat imprévisible (ce qui est la définition d'un GNA). Mais trop d'imprévu nuit.

Bref, on voit qu'un bon GNA doit fournir des résultats reproductibles : il faut pouvoir lui faire cracher la même série de valeurs dans les mêmes conditions pour pouvoir reproduire les mêmes événements.

Il doit aussi être doté d'une longue période. On appelle période d'un GNA le nombre d'appels successifs avant que ne réapparaisse la même valeur.

Le GNA que nous vous présentons dans ce volet d'introduction obéit à ces deux conditions (entre autres). Sa période est de 2 milliards et quelques, et il est parfaitement reproductible. Son petit nom est "générateur de Lehmer", du nom de l'inventeur de l'algorithme. Nous vous présenterons cet algorithme et détaillerons les notions abordées ici dans le prochain volet, car ce mois-ci, il fait trop chaud pour se serrer la tête, pas vrai?

EXEMPLE D'APPLICATION

Vous trouverez dans ce numéro le listing d'un programme en C nommé DALEKS (d'après la série "Doctor Who" de la BBC). Il provient à l'origine des gros ordinateurs dotés de terminaux, et le portage original sur ST a été effectué par Randy Eichman. C'est en raison de son origine

qu'il utilise seulement des affichages caractères (en revanche, cela le rend plus facile à porter sur une autre machine et à comprendre).

En guise de GNA, Daleks utilise un générateur de Lehmer très connu, qui a l'avantage d'être bien testé. La façon dont il est implémenté lui permet d'éviter les erreurs de dépassement de capacité arithmétique en calcul sur des entiers long (32 bits).

Pour l'utiliser, facile : on l'initialise, puis, à chaque appel, le géné renverra une valeur "aléatoire". Pour l'initialiser, deux façons : le mode normal et le mode débug. En mode normal, on passe comme argument une valeur négative, et le géné utilisera l'horloge temps réel du ST comme "semence", ou valeur initiale du géné. Par définition, celle-ci change sans arrêt et très vite. Deux lancements du programme devraient donc fournir deux valeurs différentes de l'horloge, donc de la semence. En mode débug, on passe comme argument la valeur (positive) de la semence. Cela permet de reproduire la même séquence de chiffre. Chaque valeur positive de la semence "lance" le géné sur une séquence de chiffres longue de 2 milliards. Si vous faites 2 milliards d'appels, le départ de la séquence réapparaît. Mais il faut être patient... Enfin, pour récupérer une valeur aléatoire, passer 0L comme argument. En résumé :

```
float Randm(n)
long l
```

- si n > 0L : n devient la nouvelle semence du géné;
- si n < 0L : utiliser l'horloge comme nouvelle valeur de la semence;
- si n = 0L : Randm() retourne une valeur réelle entre 0 (exclu) et 1 (exclu).

MODE D'EMPLOI DU PROGRAMME DALEKS

Daleks est un jeu simple de stratégie et de chance, dans lequel vous tentez d'échapper à des hordes de robots malfaisants mais stupides appelés DALEKS. Ceux-ci n'ont qu'un but, vous détruire. Heureusement, ils ne sont pas armés et cherchent "seulement" à vous attraper pour vous réduire en bouillie. Dans leur impatience, ils entrent en collision les uns avec les autres et vous fournissent ainsi de l'aide et des points. Plus longtemps vous leur échapperez, plus vous aurez de points. Les 10 meilleurs scores sont sauvegardés sur disque.

Le nom Daleks vient de l'émission de la BBC "Doctor Who" qui compte de nombreux fans y compris aux US où ce jeu existe en domaine public pour de nombreuses machines, y compris des "gros" utilisant des terminaux.

Vous : δ
Dalek : Ω
Débris : æ

COMMANDES CLAVIER

1-9 : Mouvements (pavé numérique)
(C)ouleurs : bascule les couleurs
(I)nstructions : affiche ces instructions
(L)aisse faire : avance automatique des robots
(S)onique : utilise le casse-robot sonique
(T)éléportation : téléporte le joueur à un endroit au hasard
e(X)it : fin de la partie; vous pouvez recommencer ou quitter le jeu.

DESCRIPTION DES COMMANDES

Mouvements : Pressez les touches 1-9 du pavé numérique pour vous déplacer dans l'une des 8 directions par rapport au "5" central. Ainsi, "1" = bas gauche, "2" = bas, "3" = bas droite, "4" = gauche, etc. Le "5" vous permet de rester immobile (et de passer votre tour). Après que vous ayez joué, les robots se déplacent à leur tour et convergent vers vous. Les mouvements sont d'une case à la fois.

Couleur : Pressez "C" pour changer les couleurs. Ceci marche en moyenne résolution (couleur) et en haute (monochrome).

Instructions : Pressez "I" pour voir les instructions à tout moment durant la partie. La partie en cours continue ensuite.

Laisse faire : Pressez "L" pour laisser les robots avancer automatiquement vers vous (comme si vous pressiez "5" en permanence). Ceci dure jusqu'à ce que vous ou tous les robots soient morts. A éviter sauf si vous êtes suicidaire, ou si vous êtes abrité derrière un débris qui anéantira tout dalek qui approcherait.

Sonique : Pressez "S" pour tirer une charge de casse-robot sonique, ce qui détruira tous les robots sur les 8 cases adjacentes à la vôtre. Cela consomme une charge. Vous gagnez une nouvelle charge au début de chaque tour. Les daleks détruits par sonique ne laissent pas de débris. Les robots se déplacent après que vous ayez tiré un sonique. La ligne d'état, en haut de l'écran, vous indique en permanence combien de charge il vous reste.

Téléportation : Pressez "T" pour activer votre téléporteur vers une case sans robot dessus ou autour (si possible). Une téléportation consomme une unité d'énergie. Les daleks se déplacent après usage du téléporteur. Vous gagnez une unité d'énergie au début de chaque tour et également lorsque vous passez sur un débris (que vous convertissez en énergie grâce à votre générateur à fusion portatif). La ligne d'état en haut de l'écran indique en permanence combien d'énergie il vous reste.

Exit : Pressez "X" pour mettre fin à la partie. Le programme vous demandera si vous voulez rejouer ou quitter le jeu.

COMMENT JOUER

1. Le jeu est divisé en tours. Vous commencez au tour 1 avec 5 unités d'énergie et une charge sonique. Vous vous déplacez le premier. Puis chacun des robots (ou daleks) se déplace. Les daleks avancent droit sur vous et il vous est recommandé d'utiliser ce comportement stupide pour les faire se heurter mutuellement. Quand deux daleks ou plus entrent en collision, leurs générateurs nucléaires se fissurent et les carcasses des daleks se changent en un tas de débris mortel pour tout robot qui le heurterait. Vous gagnez un point par robot détruit par collision ou par casse-robot sonique.

ENFIN !

S'abonner à ST Magazine ne relève plus de l'hérésie. Vous trouverez d'ailleurs ci-dessous notre nouveau bulletin d'abonnement, qui vous permet de profiter d'un tarif avantageux, et d'être tranquille pendant un an, votre magazine préféré arrivant tout droit dans votre boîte aux lettres.

NB : les abonnements en cours sont inchangés. Les tarifs précédents ne sont plus valables à la parution de ce numéro.

Bulletin d'abonnement à retourner à :

Pressimage - Abonnement ST Mag
19, rue Hégésippe Moreau
75018 PARIS

Je m'abonne à partir du numéro ..., pour un an (11 numéros) à :

➡ ST Magazine seul.

Pour ne pas manquer un seul numéro, et être ainsi au courant de toute l'actualité du ST.

- ❑ **France métropolitaine : 225 Francs**
- ❑ **Europe : 290 Francs**
- ❑ **Monde : 330 Francs**

➡ ST Magazine + disquette

Pour recevoir en plus la disquette du magazine, qui contient tous les listings publiés, ce qui évite de les taper. Les GfA Punchs obéissent à la règle. En prime, un coffret ou une reliure pour ranger les numéros de ST Magazine.

Je choisis : ❑ le coffret ❑ la reliure

- ❑ **France métropolitaine : 625 Francs**
- ❑ **Europe : 690 Francs**
- ❑ **Monde : 730 Francs**

Nom : .....
Prénom: .....
Adresse : .....
.....
CP : .....
Ville : .....

Règlement : ❑ Chèque Bancaire ❑ CCP ❑ Mandat

✧ Pour l'étranger, voyez les pages boutique ✧

Signature (des parents pour les mineurs) :

2. Vous pouvez être tué de deux manières: en étant rattrappé par un dalek, ou en vous téléportant vers une case où un dalek adjacent peut aller (très rare mais pas impossible).

3. Vous pouvez vous débarasser des daleks de 3 façons. La première (et la meilleure) façon est de les amener par des déplacements judicieux à se heurter mutuellement, laissant une pile de débris derrière laquelle vous pouvez vous abriter, ou que vous pouvez absorber. La deuxième est d'utiliser votre casse-robot sonique (mais les robots ne laissent pas de débris). La troisième est d'amener un dalek à heurter un débris.

4. Quand tous les daleks ont été éliminés, vous avancez au tour suivant, et vous gagnez une charge sonique et une unité d'énergie pour votre téléporteur. Ah oui, et aussi de nouveaux daleks furieux, 5 de plus à chaque tour.

5. Gagner des points est très simple: vous gagnez un point par robot détruit. Si votre score est supérieur à 10 lorsque vous serez finalement tué (hé hé hé...) et si vous battez un des dix meilleurs scores, votre score sera enregistré sur disque avec votre nom. N'oubliez pas d'ôter la protection en écriture de la disquette.

STRATEGIE

L'idée-force du jeu est de faire se heurter de nombreux daleks pour qu'ils laissent des piles de débris puis de vous abriter derrière. Pressez alors la touche "5" pour rester immobile tandis que les daleks avancent droit dans les débris, les sots. Quand plusieurs débris se sont formés, vous pouvez aller dansces débris pour les absorber et stocker de l'énergie pour votre téléporteur. Après quoi vous pouvez vous téléporter pour échapper aux daleks les plus proches. Quand deux daleks se rapprochent et vont former un débris, tenter de vous en rapprocher pour pouvoir absorber le débris. Plusieurs débris rapprochéspermettent de s'abriter de daleks venant de directions différentes.

NOTES

Daleks a été écrit en Mark Williams C. On ne compte plus les versions du domaine public pour différentes machines. Celle-ci a été écrite par Randy L. Eichman puis téléchargée sur les serveurs Compuserve et GEnie aux USA. Nous avons procédé au débogage et au nettoyage du source, mais ce jeu est très améliorable et assez prenant pour une courte partie entre deux compilations...

Daleks a besoin d'un fichier d'instructions (n'importe quel texte suffit à condition qu'il soit terminé par au moins un caractère de code nul) de 10K de long au maximum. Le fichier d'instructions que j'ai fait est fourni en exemple.

BILLY T.
alias Frédéric Mora.

```
/*********************************************************************/
/*      Daleks pour ST - F. Mora - Juin 1990 - Mark Williams C       */
/* N'utilise que des caractère et est donc portable sur tout système. */
/*********************************************************************/
#include <stdio.h>
#include "osbind.h"
#include "linea.h"
#include <time.h>
#define CONST_A 16807L        /* Définitions pour la fonctions Randm() */
#define CONST_M 2147483647L
#define CONST_Q 127773L
#define CONST_R 2836L
#define  TRUE          1                /* Quelques définitions utiles */
#define  FALSE         0
#define  WHO           251 /* Représente joueur dans le tableau */.
#define  DEBRIS        252 /* Représente débris dans le tableau */
#define  VIDE          0
#define  OK            0
#define  NOUVOTOUR     1
#define  MORT          2
#define  CJOUEUR       235 /* Caractère indiquant le joueur... */
#define  CROBOT        232 /* ...celui des robots... */
#define  CDEBRIS       145 /* ...et des débris */
#define  NMAXROB       99  /* Nbre max de robots sur l'écran */
#define  TLPCONSO      1 /* Energie consommée par une téléportation */
#define  ENERG0        5*TLPCONSO /* Energie au début de la partie */
#define  TAILLEINFO    7500L /* Tampon contenant les instructions */
#define  BLACK         Cconws("\33b0") /* Définitions couleurs VT-52 */
#define  RED           Cconws("\33b1")
#define  BLUE          Cconws(blue)
#define  GREEN         Cconws("\33b3")
#define  CLRHM         Cconws("\33E") /* Cclear+Home */
#define  CURSON        Cconws("\33e") /* Cursor on */
#define  CURSOFF       Cconws("\33f") /* Cursor off */
#define  REVERSVID     Cconws("\33p") /* Reverse video */
#define  NORMALVID     Cconws("\33q") /* Fin reverse */
char fname[] = "hiscore.dlk";                    /* Variables globales */
char blank[13] = "            ";/* Pour effacer un nom dans les scores*/
char *blue = "\33b2\0";
char *ecrtitre;
int score = 0,tour = 1,sonics = 1;
int coul_0,coul_1,coul_2,coul_3;
int quit = 0;
int x_robot[NMAXROB];   /* Coord. x de chaque robot */
int y_robot[NMAXROB];   /* Coord. x de chaque robot */
int ecran[39][22];      /* Tableau de l'aire de jeu */
int robots;             /* Nbre de robots restant */
int nrobs;              /* Nbre de robots au début d'un nouveau tour */
int x_who,y_who;        /* Coords x, y du joueur */
int handle,nbrnoms = 0,skores[10];
char noms[10][13],nbrlignes;
long octets;
int energie = ENERG0;   /* Energie du joueur */
int toggle = 1,rez = -1;
int crash[] = {
  0x0, 0x100, 0x200, 0x300, 0x400, 0x500, 0x600, 0x7FF, 0x800, 0x900,
  0xA00, 0xB00, 0xC00, 0xD00, 0x20, 0x103, 0x61F, 0x736, 0xBA0, 0xC0F,
  0xD00, 0x810, 0xFF00
} ;                                              /* sons Dosound() */
int kaput[] = {
  0x52, 0x103, 0x2A9, 0x301, 0x4D4, 0x500, 0x600, 0x7F8, 0x810, 0x910,
  0xA10, 0xB90, 0xC1A, 0xD00, 0xE8, 0x103, 0x2F4, 0x301, 0x4FA, 0x738,
  0x810, 0x910, 0xA10, 0xB40, 0xC1F, 0xD00, 0xFF1E, 0x7A, 0x103, 0x2BD,
  0x301, 0x4DE, 0xD00, 0xFF06, 0x52, 0x103, 0x2A9, 0x301, 0x4D4, 0xD00,
  0xFF14, 0xE8, 0x103, 0x2F4, 0x301, 0x4FA, 0xD00, 0xFF00
} ;
int sonik[] = {
  0x0, 0x100, 0x200, 0x300, 0x400, 0x500, 0x600, 0x7FF, 0x810, 0x910,
  0xA00, 0xBF4, 0xC01, 0xD0F, 0x0, 0x103, 0x2BE, 0x73C, 0x810, 0x910,
  0xBF4, 0xC01, 0xD0A, 0xFF14, 0x800, 0x900, 0xFF00
} ;
int telep1[] = {
  0x32, 0x100, 0x2C8, 0x300, 0x400, 0x500, 0x600, 0x7FC, 0x810, 0x910,
  0xA00, 0xB10, 0xC27, 0xD00, 0x101, 0x2C8, 0x301, 0x73C, 0x80C, 0x909,
  0x8000, 0x8100, 0x19FA, 0x800, 0x900, 0xFF00
} ;
int telep2[] = {
  0x32, 0x100, 0x2C8, 0x300, 0x400, 0x500, 0x600, 0x7FC, 0x810, 0x910,
  0xA00, 0xB10, 0xC27, 0xD00, 0x101, 0x2C8, 0x301, 0x73C, 0x80C, 0x909,
  0x80FA, 0x8100, 0xE700, 0x800, 0x900, 0xFF00
} ;
main()
{
  char ch;
  float Randm();
  init();                 /* Définit résol. et couleurs */
  CLRHM;
  cursxy(17, 12);
  Randm(-1L);      /* Initialise le géné avec l'horloge */
  Cconws("Voulez-vous les instructions?   (o/n)");
  ch = Crawcin();
  if ((ch == 'o') || (ch == 'O'))
    instruct();
  nouvel_ecran();         /* Efface écran, dessine bords et en-têtes */
  lit_scores();           /* Lit les meilleurs scores */
  qraph_scores();         /* Affiche "graphiques" des meilleurs scores */
  affiche_scores();       /* et affiche les chiffres des scores */
  cursxy(22, 22);
  GREEN;
  Cconws("Frappez une touche pour continuer...");
  Crawcin();
  nouvel_ecran();
  do
  {
    protagonistes();      /* Aire de jeu = joueur et tour*5 robots */
    aff_protag();         /* Affiche  robots et joueur */
    bouclejeu();
  }
  while (quit == FALSE);
  restaure();
  exit(0);
}
init()
{
  long t = 350000L;       /* Temporisation */
  if (rez < 0)            /* Seulement si 1er appel à la fonction */
  {
    coul_0 = Setcolor(0, -1);/* Sauvegarde anciennes couleurs */
    coul_1 = Setcolor(1, -1);
    coul_2 = Setcolor(2, -1);
    coul_3 = Setcolor(3, -1);
    linea0();
    hidemouse();          /* Cache souris */
    CURSOFF;              /* Cache curseur */
```

```
 }
 rez = Getrez();
 if (rez == 2)          /* Si haute résolution */
 {
   Setcolor(0, 0x000); /* Couleur 0 = noir */
   Setcolor(1, 0x777); /* 1 = Blanc */
   Setcolor(2, 0x777); /* 2 = Blanc */
   Setcolor(3, 0x777); /* 3 = Blanc */
   blue = "\33b1\0";   /* Bleu devient la couleur 1 */
 }
 else    /* Sinon établit couleurs moyenne rez. */
 {
   Setcolor(0, 0x000); /* Couleur 0 = noir */
   Setcolor(1, 0x700); /* Rouge */
   Setcolor(2, 0x047); /* Bleu */
   Setcolor(3, 0x060); /* Vert */
 }
 if (rez == 0) /* Si basse rez. demande de changer la résolution */
 {
   CLRHM;
   BLUE;
   cursxy(6, 10);
   Cconws("Daleks ne marche qu'en");
   cursxy(7, 12);
   Cconws("moyenne ou haute résolution");
   while (t > 0)       /* Temporisation (très laid) */
     -t;
   restaure();
   exit(1);
 }
}
instruct()
{
  int x = 0,nbrlign = 0;
  static char info[TAILLEINFO+2];
  char ch = 1;
  int page1 = 0;                    /* Flag 1ère page */
  if (info[0] == 0)     /* Si instructions pas chargées, les lire */
  {
    handle = Fopen("info.dlk", 0);
    Fread(handle, TAILLEINFO, info);
    Fclose(handle);
  }
  CLRHM;
  BLUE;
  while ((ch = info[x]) != 0)/* Afficher caractères  jusqu'à un '\0' */
  {
    Cconout(ch);
    ++x;
    if (ch == '\n')     /* Compte les \n, pause toutes les 21 lignes */
    {
      ++nbrlign;
      if (nbrlign >= 21)
      {
        if (page1 == 0)     /* Si 1ère page, affiche les symboles */
        {
          page1 = 1;
          cursxy(69, 14);
          REVERSVID;
          Cconout(CJOUEUR); /* Note de F. Mora : cette */
          NORMALVID;        /* méthode d'affichage des */
          cursxy(69, 16);   /* symboles est horrible, */
          RED;              /* mais elle marche, alors */
          Cconout(CROBOT);  /* je la laisse (et tout le */
          cursxy(69, 18);   /* source original était du */
          GREEN;            /* même tonneau!) */
          Cconout(CDEBRIS);
        }
        nbrlign = 0;
        cursxy(20, 24);
        GREEN;
        Cconws("X pour sortir, autre touche pour suite");
        ch = Crawcin();
        if ((ch == 'x') || (ch == 'X'))
          return ;
        CLRHM;
        BLUE;
      }
    }
  }
  cursxy(15, 24);
  GREEN;
  Cconws("Frapez une touche pour continuer (dernière page)");   Craw-
cin();
}
nouvel_ecran()
{
  int ligne;
  CLRHM;      /* Efface l'écran, affiche ligne d'état */
  GREEN;
  Cconws("Score:");
  cursxy(23, 0);
  Cconws("Tour:");
  cursxy(40, 0);
  Cconws("Sonics:");
  cursxy(54, 0);
  Cconws("Energie:");
  GREEN;      /* Affiche valeurs courants dans ligne d'état */
  cursxy(7, 0);
  print(score);
  cursxy(29, 0);
  print(tour);
  cursxy(48, 0);
  print(sonics);
  cursxy(63, 0);
  print(energie);
  BLUE;       /* Affiche les bordures haute et basse */
  cursxy(1, 1);
  Cconws("________________________________");
  Cconws("________________________________");
  Cconout('_');
  cursxy(1, 24);
  Cconws("----------------");
  Cconws("----------------");
  Cconout('-');
  for (ligne = 2; ligne < 24; ligne++)
  {           /* Affiche les bordures droite et gauche */
    cursxy(0, ligne);
    Cconout('|');
    cursxy(78, ligne);
    Cconout('|');
  }
}
cursxy(x,y)  /* Met le curseur en colonne x et ligne y */
```

```
  int x,y;
{
  Cconws("\33Y"); /* Code escape VT-52 de positionnement du curseur */
  Cconout(y+32);
  Cconout(x+32);  /* D'abord x puis y séparés par des espaces */
}
print(num)                /* Affiche une variable entière */
  int num;
{
  int q,i;
  short r;
  short t[8];
  i = 0;
  do
  {
    q = num/10;
    r = num-10*q;
    t[i++] = r;
    num = q;
  }
  while (num != 0);
  do
  {   /* t[] contient la décomposition en chiffres de num.  Affiche */
    Bconout(5, 16+t[-i]);
  }
  while (i != 0);
  Bconout(5, 32); /* Met un espace à la fin du dernier chiffre */
}
lit_scores()
{
  int n,nouveau;
  long t = 100000L;
  BLUE;    /* Ouvre le fichier des hiscrores ou en crée un */
  do
  {

    handle = Fopen(fname, 2);/* Tente d'ouvriren lecture/écriture */
    if (handle < 0)      /* S'il n'existe pas... */
    {
      handle = Fcreate(fname, 0);/* tente de le créer */
      if (handle >= 0)  /* Si la création aboutit */
      {
        nouveau = TRUE;
        nouvel_ecran();
      }
    }
    else
      nouveau = FALSE;
  }
  while (handle < 0);/* Essaie tant que ça ne marche pas */
  if (nouveau == FALSE) /* Charge les meilleurs scores en mémoire */
  {
    octets = Fread(handle, 1L, &nbrlignes);
    if (octets > 0)
    {
      nbrnoms = (int)nbrlignes;
      for (n = 0; n < nbrnoms; n++)
        Fread(handle, 12L, noms[n]);
      for (n = 0; n < nbrnoms; n++)
        Fread(handle, 2L, &skores[n]);
    }
  }
  Fclose(handle);
}                         /* Fin de lit_scores() */
graph_scores()          /* Affiche les "graphiques" des hiscores */
{
  int n;
  cursxy(27, 4);
  for (n = 1; n < 5; n++)
  {
    GREEN;
    Cconout('*');
    Cconout(' ');
    RED;
    Cconout('*');
    Cconout(' ');
    BLUE;
    Cconout('*');
    Cconout(' ');
  }
  GREEN;
  Cconout('*');
  cursxy(27, 21);
  for (n = 1; n < 5; n++)
  {
    GREEN;
    Cconout('*');
    Cconout(' ');
    RED;
    Cconout('*');
    Cconout(' ');
    BLUE;
    Cconout('*');
    Cconout(' ');
  }
  GREEN;
  Cconout('*');
  for (n = 5; n < 20; n += 3)
  {
    RED;
    cursxy(26, n);
    Cconout('*');
    cursxy(52, n);
    Cconout('*');
    BLUE;
    cursxy(26, n+1);
    Cconout('*');
    cursxy(52, n+1);
    Cconout('*');
    GREEN;
    cursxy(26, n+2);
    Cconout('*');
    cursxy(52, n+2);
    Cconout('*');
  }
  RED;
  cursxy(26, 20);
  Cconout('*');
  cursxy(52, 20);
  Cconout('*');
  GREEN;
  cursxy(30, 6);
  Cconws("T O P   S C O R E S");
  BLUE;
```

```
  cursxy(29, 9);
  Bconout(5, 17);
  Cconout('.');          /* Numéro du hi score */
  for (n = 11; n < 19; n++)
  {
    cursxy(29, n);
    Bconout(5, n+7);  /* Conversion en ASCII du numéro */
    Cconout('.');
  }
  cursxy(28, 19);
  Bconout(5, 17);
  Bconout(5, 16);
  Cconout('.');
}                        /* Fin de graph_scores() */
affiche_scores()       /* Affiche noms et scores */
{
  int n;
  BLUE;
  if (nbrnoms > 0)
  {
    cursxy(32, 9);
    Cconws(noms[0]);
    cursxy(45, 9);
    print(skores[0]);
    for (n = 1; n < nbrnoms; n++)
    {
      cursxy(32, n+10);
      Cconws(noms[n]);
      cursxy(45, n+10);
      print(skores[n]);
    }
  }
}                        /* Fin de affiche_scores() */
protagonistes()   /* Remplit l'aire de jeu avec robots et joueur */
{
  int x,y,n = 1;
  nrobs = tour *5;
  if (nrobs > NMAXROB)  /* nrobs = minimum (tour * 5, NMAXROB) */
    nrobs = NMAXROB;
  robots = nrobs;        /* Nbre robots restant = effectif de départ */
  while (n <= nrobs)     /* Il faut placer n robots */
  {
    x = Equiprobable(0,38); /* Coord x entre 0 et 38 */
    y = Equiprobable(0,21); /* Coord y entre 0 et 21 */
    if (ecran[x][y] == VIDE)     /* Il faut que la case soit vide */
    {
      x_robot[n] = x;
      y_robot[n] = y;
      ecran[x][y] = n;  /* Et un robot de casé, un! */
      n++;
    }
  }
  x = FALSE;
  while (x == FALSE)
  {       /* Placer le joueur dans une case vide */
    x_who = Equiprobable(0, 38); /* entre 0 et 38 */
    y_who = Equiprobable(0, 21);./* entre 0 et 21 */
    if (ecran[x_who][y_who] == VIDE)
    {
      ecran[x_who][y_who] = WHO;
      x = TRUE;
    }
  }
}                        /* Fin de protagonistes() */
int Equiprobable(a, b)    /* Retourne un entier équiprobable */
  int a,b;
{   float Randm();
    return ((int)((b-a) * Randm(0L) + a));
}
aff_protag()     /* Affichage des protagonistes */
{
  int x,y,n;
  for (n = 1; n <= nrobs; n++)/* Affiche les robots */
  {
    if (x_robot[n] < 100)/* Si pas débris ou mort */
    { /* Affiche le robot à sa coord */
      cursxy((x_robot[n]*2)+1, y_robot[n]+2);
      RED;
      Cconout(CROBOT);     /* Affiche le car. du robot */
    }
    else
      if (x_robot[n] < 200)/* Sinon, est-ce un débris? */
      {
        x = x_robot[n]-100;/* La coord x est calculée */
        y = y_robot[n];
        cursxy((x *2)+1, y+2);
        GREEN;
        Cconout(CDEBRIS); /* Affiche le car. du débris */
      }
  }
  cursxy((x_who *2)+1, y_who+2);/* Affiche le joueur... */
  BLUE;
  REVERSVID;                    /* ...en vidéo inverse */
  Cconout(CJOUEUR);
  NORMALVID;
}
bouclejeu() /* Traite l'action du joueur */
{
  int action,status = OK,flag,cmpteur;
  int xw,yw;
  while (status == OK)
  {
    while (Cconis() != 0)/* Vide le buffer clavier */
      Crawcin();
    action = Crawcin();  /* Lit une touche au clavier */
    switch (action)
    {
      case '1' :         /* Direction 1 (bas, gauche) */
        status = depl_joueur(-1, 1);
        break;
      case '2' :         /* Direction 2 (bas) */
        status = depl_joueur(0, 1);
        break;
      case '3' :         /* Direction 3 (bas, droite) */
        status = depl_joueur(1, 1);
        break;
      case '4' :         /* Direction 4 (gauche) */
        status = depl_joueur(-1, 0);
        break;
      case '5' :         /* Ne bouge pas le joueur */
        status = depl_robots();/* Les robots bougent */
        break;
      case '6' :         /* Direction 6 (droite) */
        status = depl_joueur(1, 0);
```

```
  break;
case '7' :          /* Direction 7 (haut, gauche) */
  status = depl_joueur(-1, -1);
  break;
case '8' :          /* Direction 8 (haut) */
  status = depl_joueur(0, -1);
  break;
case '9' :          /* Direction 9 (haut, droite) */
  status = depl_joueur(1, -1);
  break;
case 'c' :          /* Inverse les couleurs */
case 'C' :
  if (toggle == 1)
  {
    if (rez == 1)
    {
      Setcolor(0, 0x642);
      Setcolor(1, 0x000);
      Setcolor(2, 0x000);
      Setcolor(3, 0x000);
    }
    else
    {
      Setcolor(0, 0x777);
      Setcolor(1, 0x000);
      Setcolor(2, 0x000);
      Setcolor(3, 0x000);
    }
  }
  else
    init();
  toggle = -toggle;
  break;
case 'i' :          /* Affiche les instructions */
case 'I' :
  instruct();
  nouvel_ecran();
  aff_protag();
  break;
case 'l' :          /* Laisse les robots avancer */
case 'L' :
  while (status == OK)
    status = depl_robots();
  break;
case 's' :          /* Tire une charge de Sonic */
case 'S' :
  if (sonics > 0) /* Reste-t-il des charges ? */
  {
    sonics -= 1;
    cursxy(48, 0); /* Décrémente les charges */
    GREEN;          /* Et met à jour la ligne d'état */
    print(sonics);
    status = detr_robot(-1, 1);/*Détruit dans les 8 cases autour*/
    if (status == NOUVOTOUR)/* Si dernier robot, */
      break;                /* nouveau tour */
    status = detr_robot(0, 1);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(1, 1);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(-1, 0);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(1, 0);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(-1, -1);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(0, -1);
    if (status == NOUVOTOUR)
      break;
    status = detr_robot(1, -1);
    if (status == NOUVOTOUR)
      break;
    status = depl_robots(); /* C'est aux robots de se déplacer */
  }
  break;
case 't' :          /* Téléportation */
case 'T' :
  if (energie >= TLPCONSO) /* Si assez d'énergie */
  {
    energie -= TLPCONSO;/* Soustrait la conso du téléporteur */
    cursxy(63, 0); /* et met à jour la ligne d'état */
    GREEN;
    print(energie);
     ecran[x_who][y_who] = VIDE; /* Efface le joueur de sa case */
    Dosound(telep1);     /* 1ère partie du son */
    delai();             /* attend un peu */
    cursxy((x_who *2)+1, y_who+2);
    Cconout(' ');  /* Efface le joueur à l'écran */
    flag = FALSE; /* Tente de trouver une case vide sans robot */
    cmpteur = 0;
    xw = Equiprobable(1,37);/* X départ de 1 à 37 */
    yw = Equiprobable(1,20);/* Y départ de 1 à 20 */
    do
    {
      if (case_ok(xw, yw))
      {
        flag = TRUE;
        break;
      }          /* Si case pas OK, essaie (xw, yw) adjacent */
      cmpteur++;
      xw++;
      if (xw > 37) /* Si fin de ligne, aller à la ligne suivante*/
      {
        xw = 1;
        yw++;
        if (yw > 20)/* Si dernière ligne, aller à */
          yw = 1;   /* la première ligne */
      }
    }
    while (cmpteur < 740);/* Fait 37 * 20 = 740 tentatives max */
    if (flag == TRUE)              /* Si case OK, */
    {
      x_who = xw;  /* placer le joueur dessus */
      y_who = yw;
    }
    else /* Sinon le mettre sur une case vide */
      do /* quelconque */
      {
        x_who = Equiprobable(0,38);
        y_who = Equiprobable(0,21);
```

```
        }
        while (ecran[x_who][y_who] != VIDE);
        ecran[x_who][y_who] = WHO; /* Met joueur dans l'aire de jeu */
        cursxy((x_who *2)+1, y_who+2);
        Dosound(telep2);
        delai();
        BLUE;
        REVERSVID;
        Cconout(CJOUEUR);          /* Et affiche joueur */
        NORMALVID;                 /* en vidéo inverse */
        status = depl_robots(); /* Les robots bougent */
      }            /* Fin if (energy >= TLPCONSO) */
      else         /* Si pas assez d'énergie, */
      {        /* le téléporteur claque à vide */
        Dosound(crash);
        delai;
      }
      break;
    case 'x' :         /* Quitte le jeu */
    case 'X' :
      status = MORT;
      break;
  }                    /* Fin switch */
}                      /* Fin while */
if (status == NOUVOTOUR)
{
  eff_aire_jeu();
  tour++;
  sonics++; /* 1 charge Sonic et 1 charge téléporteur
               à chaque nouveau tour */
  energie += TLPCONSO;
  nouvel_ecran();
}
else
  if (status == MORT)
  {
    Dosound(kaput);
    if (score > 10)
      enreg_score();
    eff_aire_jeu();
    tour = 1;
    sonics = 1;
    score = 0;
    energie = ENERGO;
    cursxy(26, 1);
    GREEN;
    Cconws("Voulez-vous rejouer? (o/n)");
    flag = FALSE;
    while (flag == FALSE)/* Lit une touche jusqu'à 'o' ou 'n' */
    {
      action = Crawcin();
      switch (action)
      {
        case 'n' :
        case 'N' :
          quit = TRUE;
          flag = TRUE;
          break;
        case 'o' :
        case 'O' :
          nouvel_ecran();
          flag = TRUE;
```

```
        }
      }
    }
}
int case_ok(xw,yw) /* Vérifie que (xw, yw) et les 8 cases autour vides*/
  int xw,yw;
{
  register int x,y,ok;
  ok = TRUE;
  for (x = -1; x < 2; ++x)
  {
    for (y = -1; y < 2; ++y)
    {
      ok = ok && (ecran[x + xw][y + yw] == VIDE);
      if (ok == FALSE)
        break;
    }
    if (ok == FALSE)
      break;
  }
  return (ok);
}                          /* Fin de case_ok() */
delai()  /* Temporisation pour laisser les sons finir */
{
  long z = 25000L;
  while (z > 0)
    z -= 1;
}
depl_joueur(x,y)   /* Déplace joueur dans direction indiquée par x, y */
  int x,y;
{
  int status;
  ecran[x_who][y_who] = VIDE; /* Efface ancienne case ... */
  cursxy((x_who *2)+1, y_who+2); /* ...et à l'écran */
  Cconout(' ');
  x_who += x;          /* Ajoute déplacement voulu sur x */
  if ((x_who < 0) || (x_who > 38)) /*Vérifie nouvel x dans les limites*/
    x_who -= x;          /* Si non, reste à l'ancien x */
  y_who += y;          /* Ajoute déplacement voulu sur y */
  if ((y_who < 0) || (y_who > 21)). /*Vérifie nouvel y dans les limites*/
    y_who -= y;          /* Si non, reste à l'ancien y */
  if (ecran[x_who][y_who] == DEBRIS) /*S'il y a des débris sur la case*/
  {
    energie++;           /* Absorbe le débris */
    ecran[x_who][y_who] = VIDE;
    cursxy(63, 0);       /* Met ligne d'état à jour */
    GREEN;
    print(energie);
  }  /* Met le curseur à la nouvelle position du joueur */
  cursxy((x_who *2)+1, y_who+2);
  if (ecran[x_who][y_who] == VIDE)   /* Si case vide, */
  {
    BLUE;
    REVERSVID;  /* affiche le joueur en vidéo inverse */
    Cconout(CJOUEUR);
    NORMALVID;
    ecran[x_who][y_who] = WHO;/* Met joueur dans A.J. */
    status = depl_robots();/* Fait bouger les robots */
  }  /* Si nouvelle case pas vide, le joueur est mort */
  else
  {
    BLUE;
```

```
    Cconout('*');
    status = MORT;
  }
  return (status);
}
depl_robots() /* Effectue les déplacement des robots */
{
  int x,y,n;
  for (n = 1; n <= nrobs; n++)     /* Pour chaque robot */
  {                          /* S'il a été détruit, l'effacer */
    if (x_robot[n] < 100)
      ecran[x_robot[n]][y_robot[n]] = VIDE;
  } /* Calcule le déplacement de chaque robot (pour se rapprocher) */
  for (n = 1; n <= (nrobs); n++)
  {
    if (x_robot[n] < 100)       /* Si robot non détruit */
    {
      if (x_robot[n] < x_who)  /* Déplacement sur x */
        x = x_robot[n]+1;
      else
        if (x_robot[n] > x_who)
          x = x_robot[n]-1;
        else
          x = x_robot[n];
      if (y_robot[n] < y_who) /* Déplacement sur y */
        y = y_robot[n]+1;
      else
        if (y_robot[n] > y_who)
          y = y_robot[n]-1;
        else
          y = y_robot[n];
      switch (ecran[x][y])          /* Le robot a-t-il heurté qqchose ? */
      {
        case  VIDE :
          if (ecran[x_robot[n]][y_robot[n]] == VIDE)
          {    /* Efface robot si necéssaire */
            cursxy((x_robot[n]*2)+1, y_robot[n]+2);
            Cconout(' ');
          }
          x_robot[n] = x;
          y_robot[n] = y;     /* Change position robot */
          ecran[x][y] = n;/* Met numéro robot dans A.J. */
          cursxy((x *2)+1, y+2);
          RED;
          Cconout(CROBOT);    /* Affiche robot à nouvelle position */
          break;
        case  DEBRIS :
          if (ecran[x_robot[n]][y_robot[n]] == VIDE)
          { /* Efface ancienne case si nécessaire */
            cursxy((x_robot[n]*2)+1, y_robot[n]+2);
            Cconout(' ');
          }
          x_robot[n] += 200;/* Si débris, robot foutu */
          score++;            /* +1 au score */
          Dosound(crash);
          delai();
          cursxy(7, 0);
          GREEN;
          print(score);       /* Met score à jour */
          robots -= 1;
          if (robots == 0)/* Si dernier robot, nouveau tour */
            return (NOUVOTOUR);
          break;
        case  WHO :   /* Si le robot atteint le joueur */
          if (ecran[x_robot[n]][y_robot[n]] == VIDE)
          { /* Efface case si nécessaire */
            cursxy((x_robot[n]*2)+1, y_robot[n]+2);
            Cconout(' ');
          }
          cursxy((x *2)+1, y+2);
          BLUE;
          Cconout('*');
          return (MORT);
          break;
        default  : /* Sinon, c'est une collision entre robots */
          x_robot[ecran[x][y]] += 100;/* Change ce robot en débris */
          ecran[x][y] = DEBRIS;/* Met débris dans A.J. */
          if (ecran[x_robot[n]][y_robot[n]] == VIDE)
          {   /* Efface case si nécessaire */
            cursxy((x_robot[n]*2)+1, y_robot[n]+2);
            Cconout(' ');
          }
          x_robot[n] += 200;   /* Ce robot est détruit */
          cursxy((x *2)+1, y+2);
          GREEN;
          Cconout(CDEBRIS);          /* Affiche débris */
          score += 2;                /* +2 au score */
          Dosound(crash);
          delai();
          cursxy(7, 0);
          print(score);          /* Met à jour le score */
          robots -= 2;
          if (robots == 0)          /* Si dernier robot, */
            return (NOUVOTOUR);   /* nouveau tour */
      }                    /* Fin de switch */
    }                      /* Fin de if (robot pas détruit) */
  }                        /* Fin de for */
  return (OK);
}                          /* Fin de depl_robot */
detr_robot(x,y)            /* Détruit robot par Sonic */
  int x,y;   /* x, y = déplacement relatif au joueur */
{
  int xflag = 0,yflag = 0,n,status = OK; /*Vérifie x dans les limites */
  if ((x_who+x >= 0) && (x_who+x <= 38))
    xflag = TRUE;              /* Vérifie que y est dans les limites */
  if ((y_who+y >= 0) && (y_who+y <= 21))
    yflag = TRUE;
  if (xflag && yflag)      /* Si les deux sont bons */
  {
    n = ecran[x_who+x][y_who+y];
    if ((n > 0) && (n < WHO))      /* Est-ce un robot ? */
    {
      Dosound(sonik);
      delai();
      ecran[x_robot[n]][y_robot[n]] = VIDE; /* Efface robot détruit */
      cursxy((x_robot[n]*2)+1, y_robot[n]+2);
      Cconout(' ');
      x_robot[n] += 200;/* Marque le robot comme étant détruit */
      score++;            /* +1 au score */
      GREEN;
      cursxy(7, 0);
      print(score);       /* Met le score à jour */
      robots -= 1;   /* Si dernier robot, nouveau tour */
      if (robots == 0)
```

```
        status = NOUVOTOUR;
      }
    }
    return (status);
} /* Fin de detr_robot */
eff_aire_jeu()                          /* Nettoie l'A.J. */
{
  int n;
  for (n = 1; n <= nrobs; n++)
  {
    if (x_robot[n] < 100)
      ecran[x_robot[n]][y_robot[n]] = VIDE;
    else
      if (x_robot[n] < 200)
        ecran[x_robot[n]-100][y_robot[n]] = VIDE;
  }
  ecran[x_who][y_who] = VIDE;
}
enreg_score() /* Enregistre nom et score d'un champion */
{
  int n = 0,flag = FALSE,x;
  char ch;
  while ((flag == FALSE) && (n < 10)) /* Trouve le plus bas score */
  {
    if (score >= skores[n])/* Score courant plus haut? */
      flag = TRUE;          /* Si oui, sortie; n = rang */
    else
      ++n;
  }
  if (n < 10) /* Si n<10, score = le plus mauvais sinon, son rang est */
  {           /* de 0 à 9. Décale les scores d'un cran */
    for (x = 8; x >= n; x--)
    {
      strcpy(noms[x+1], noms[x]); /* Les noms... */
      skores[x+1] = skores[x];    /* ...et les scores */
    }
    skores[n] = score;          /* Sauve score courant */
    strcpy(noms[n], blank);     /* Efface ancien nom */
    if (nbrnoms < 10) /* Si tableau pas plein, incrémente nb de scores*/
      ++nbrnoms;
    cursxy(22, 1);
    GREEN;
    Cconws("Frappez une touche pour continuer...");
    Crawcin();
    nouvel_ecran();
    graph_scores();             /* Affiche les hi scores */
    affiche_scores();
    x = 0;
    if (n == 0)   /* Positionne le curseur pour la frappe du nom */
      cursxy(32, 9);    /* Position pour score no. 1 */
    else
      cursxy(32, n+10); /* Position pour autre score */
    CURSON;
    BLUE;
    while (x < 12)      /* 12 cars. maxi pour le nom */
    {
      ch = Cconin();/*Lit un car au clavier, Si pas backspace ou Enter*/
      if ((ch != 13) && (ch != 8))
      {
        noms[n][x] = ch; /* Met le car lu dans le nom */
        ++x;             /* Passe au car suivant */
      }
      else
        if (ch == 8)    /* Si backspace, décrémente x */
        {
          if (x != 0)   /* sauf si début de ligne */
            -x;
          else
            Cconout(' ');/* Si début ligne,met un blanc */
        }
        else
          x = 12;       /* Si Enter, sortie de boucle */
    }
    CURSOFF;            /* Eteint le curseur */
    ecrit_hiscores();   /* Ecrit le tableau des scores sur disque */
    cursxy(22, 22);
    BLUE;
    Cconws("Frappez une touche pour continuer...");
    Crawcin();
    nouvel_ecran();
  }                     /* Fin de if n < 10 */
}                       /* Fin de enreg_score() */
ecrit_hiscores()
{
  int n;
  nbrlignes = (char)nbrnoms;
  handle = Fopen(fname, 2);
  Fwrite(handle, 1L, &nbrlignes);
  for (n = 0; n < nbrnoms; n++)
    Fwrite(handle, 12L, noms[n]);
  for (n = 0; n < nbrnoms; n++)
    Fwrite(handle, 2L, &skores[n]);
  Fclose(handle);
}                       /* Fin de ecrit_hiscores() */
restaure()              /* Restaure couleurs du bureau */
{
  Setcolor(0, coul_0);
  Setcolor(1, coul_1);
  Setcolor(2, coul_2);
  Setcolor(3, coul_3);
  GREEN;                /* Remet le curseur en couleur 3 (défaut) */
  CLRHM;                /* Efface écran */
  showmouse();          /* Montre la souris */
}
float Randm(l) long l;
{ /* Retourne un float entre 0 et 1 exclus */
  static unsigned long s;
  long t;
  if (l == 0L)   /* Cas général, on veut une valeur */
  {      /* % est l'opérateur modulo */
    t = CONST_A * (s % CONST_Q) -      CONST_R * (long)(s / CONST_Q);
    if (t > 0L)
      s = t;
    else
      s = t + CONST_M;
    return((float)s / (float) CONST_M);
  }
  else  /* On veut initialiser le générateur, si init automatique, */
    if (l < 0L) /* prendre l'horloge système comme semence */
      s = (unsigned long) clock();
    else        /* Sinon on utilise la valeur >0 passée en argument */
      s = (unsigned long) l;
    return(0.0);
} /* Fin Randm(); */
```

INITIATION A L'ASSEMBLEUR (XVI)

Après la pause du mois dernier, avec la présentation sommaire de notre outil de développement, il est grand temps de réactiver nos neurones en reprenant la suite de notre étude détaillée des instructions. Pour cette période de vacances, nous abordons l'instruction "MOVEP" et l'adressage des périphériques, histoire de bronzer de l'intérieur.

Si vous vous souvenez, il y a très longtemps, nous avions parlé de "bits", en soulignant que le 68000 était un microprocesseur 16 bits. Néanmoins, ce dernier est capable d'adresser des circuits périphériques 8 bits en taille octet (.B). L'instruction MOVEP permet d'adresser ces mêmes périphériques 8 bits selon un format mot (.W) ou long-mot (.L).

Tout d'abord, il faut savoir que les registres d'un circuit périphérique 8 bits ne peuvent se situer, dans une configuration mémoire 68000 qu'à des adresses exclusivement paires ou impaires, comme le montre le schéma 15.1 sur lequel on considère un périphérique classique constitué de six registres REG1 à REG6.

Le pourquoi de cette sombre histoire d'adressage dépasse le cadre de la présente initiation (ouf ! je l'ai échappé belle). Néanmoins, les plus curieux pourront toujours essayer de chercher sur le 68000 une broche A0 (sur le bus d'adresse) et se pencher sur la connexion matérielle entre le micro et le périphérique...

Pour bien comprendre l'intérêt de l'instruction MOVEP, considérons l'initialisation des six registres du pseudo périphérique précédent avec les valeurs suivantes :
REG5 = $00, REG6 = $FF, REG1 = $FA, REG2 = $DE, REG3 = $A2, REG4 = $C1 (selon l'ordre précisé).
On peut réaliser cette initialisation avec la classique instruction MOVE en adressant des octets. Supposons que l'adresse de base de ce périphérique soit égale à $000FA000 et qu'il soit situé aux adresses impaires.

On peut alors définir les équivalences suivantes :

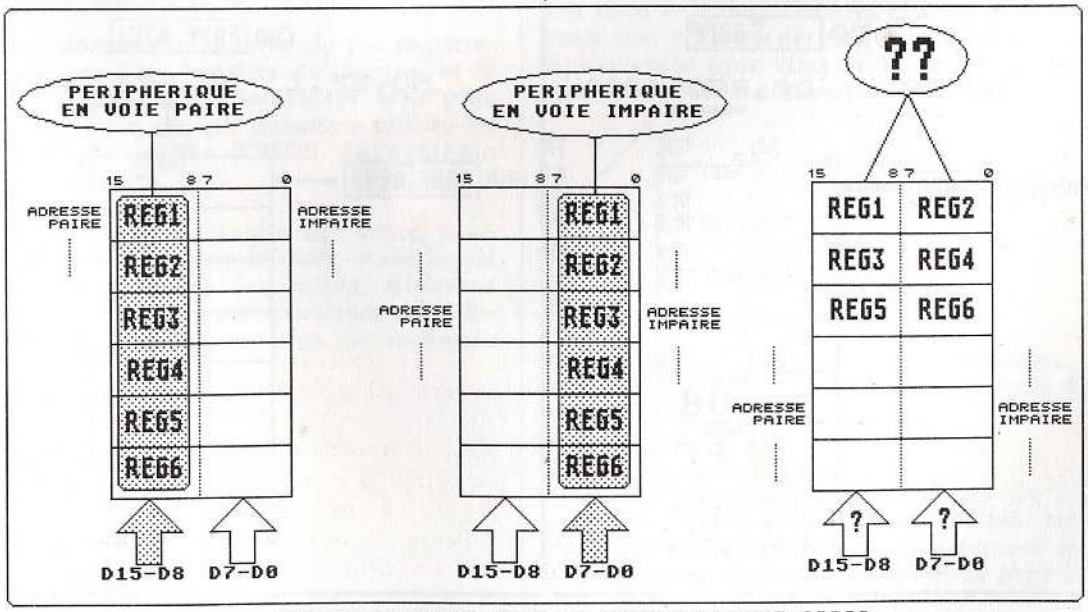
```
ADRESS_IO EQU   $000FA000    /* adresse de base */
REG1      EQU   1            /* 1 octet par rapport à ... */
REG2      EQU   3            /* 3 octet par rapport à ... */
REG3      EQU   5            /* ... */
REG4      EQU   7
REG5      EQU   9
REG6      EQU   11
```

Les REGx sont définis tels des offsets par rapport à l'adresse de base du circuit, déterminée par ADRESS_IO. Le circuit étant réellement aux adresses impaires, l'adresse "de base" ADRESS_IO étant déclarée paire, les déplacements REGx sont forcément impairs... Logique ! non ?
Ce type de définition d'équivalence permet d'utiliser aisément l'adressage *offset_16_bits(registre_Adresse)* :

```
LEA     ADRESS_IO,A0
MOVE.B  #$00,REG4(A0)
MOVE.B  #$FF,REG5(A0)
MOVE.B  #$FA,REG0(A0)
MOVE.B  #$DE,REG1(A0)
MOVE.B  #$A2,REG2(A0)
MOVE.B  #$C1,REG3(A0)
```

Résumons : s'il est aussi facile d'initialiser notre périphérique avec des MOVE.B classiques, l'instruction MOVEP n'est qu'une fumisterie ? Diantre que neni! (pour les moyenâgeux), ben non! (pour les autres).
Lorsqu'il s'agit d'un périphérique possédant plus d'une vingtaine de registres (si, ça existe, le MFP en possède 24), on apprécie de pouvoir les initialiser deux par deux ou même quatre par quatre... c'est-à-dire selon un format MOT ou LONG-MOT avec l'instruction MOVEP.
Les seules possibilités d'adressage autorisées avec le MOVEP sont :

```
MOVEP.W   Dn,DEP16(An)
MOVEP.L   Dn,DEP16(An)
```



SCHEMA 15.1 PERIPHERIQUE 8 BITS EN ENVIRONNEMENT 68000

pour la programmation (écriture) des registres d'un périphérique, et :

```
MOVEP.W   DEP16(An),Dn
MOVEP.L   DEP16(An),Dn
```

pour la lecture des registres.

L'utilisation exclusive du mode d'adresage DEP16(An) explique que les équivalences de définition des registres d'un périphérique sont faits relativement à une adresse de base (avec laquelle on initialise évidemment le registre d'adresse An. Ainsi l'exemple de programmation précédent avec les MOVE.B se résume à

```
LEA       ADRESS_IO,A0
MOVE.W    #$00FF,d0
MOVEP.W   D0,REG4(A0)
MOVE.L    #$FADEA2C1,D0
MOVEP.L   D0,REG0(A0)
```

Le schéma 15.2 illustre cet exemple de transfert.

Mais au fait, une question à 10 bits : un transfert (vers un périphérique 8 bits) sous forme MOT ou LONG-MOT avec un MOVE, ça donne quoi ?
En supposant notre périphérique aux adresses impaires, deux cas peuvent se présenter :

- une erreur d'adresse lors d'un transfert MOT ou LONG-MOT à une adresse IMPAIRE;
- une erreur bus si les cases mémoire paires, en regard des cases impaires où se situe le périphérique, ne sont pas reconnues par le décodage matériel.

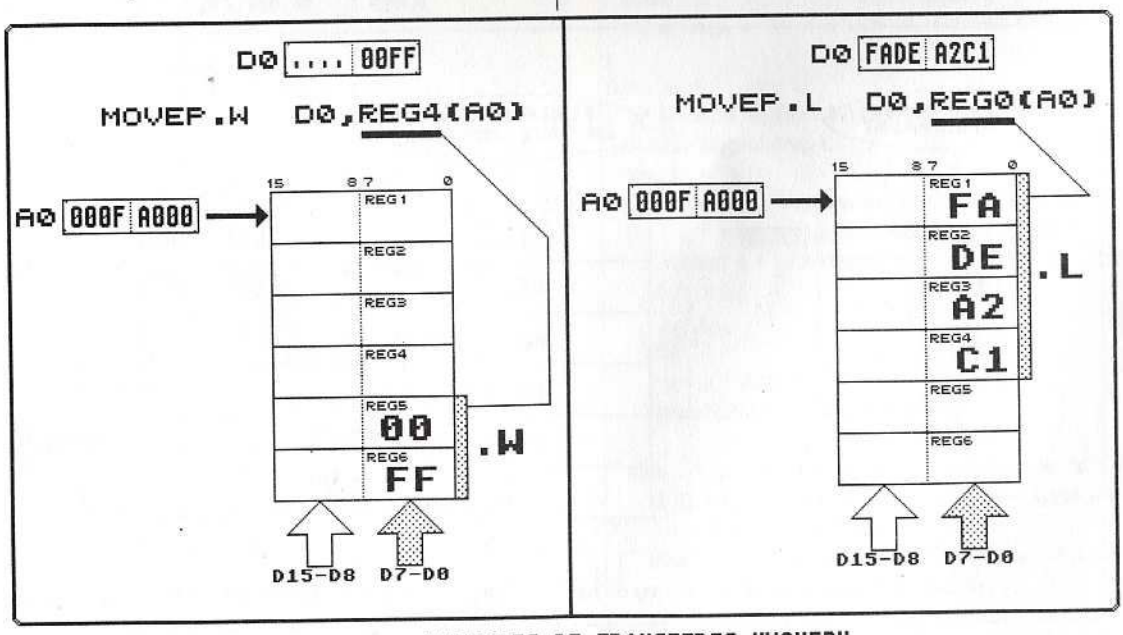
Avant d'aborder la "map" des périphériques sur un ST (cas d'un 1040 STF), revenons sur les possibilités de fonctionnement du 68000 en mode utilisateur et superviseur...
Dans le registre d'état du microprocesseur (le SR), le bit S reflète le mode de fonctionnement : S=1 le micro est en mode superviseur, S=0 il est en mode utilisateur. Quelquefois, pour accéder aux zones mémoire réservées, votre programme devra s'exécuter en mode superviseur... Heureusement, le système d'exploitation intègre une fonction permettant de faire exécuter une routine en mode superviseur en lui passant en paramètre l'adresse de la routine. Un appel typique de cette fonction vous est proposé ci-dessous :

```
Supexec   equ      $26

          pea.l    adresse          paramètre dans la pile
          move.w   #Supexec,-(a7)   code de la fonction
          trap     #14              appel XBIOS (On verra)
          addq.l   #6,a7            réalignement de pile

adresse   equ      *
          ...  Début routine à exécuter en mode superviseur
```

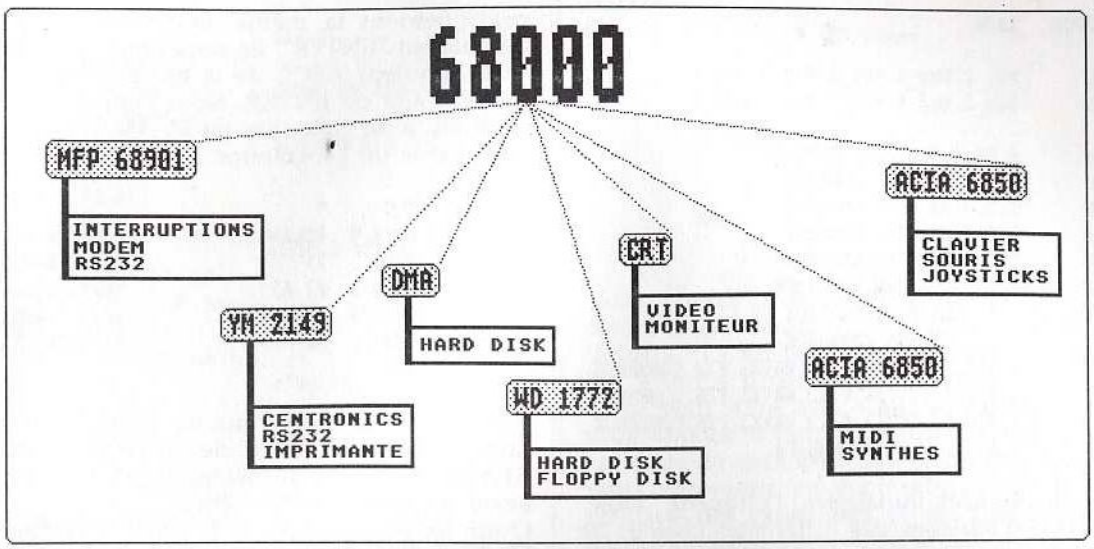
Avec cette fonction, vous allez (enfin ?) pouvoir bidouiller dans des zones encore inexplorées, telle que la zone périphérique superviseur ou celle des variables globales du système d'exploitation... Aie ! Attendez au moins la rentrée qu'on vous explique à quoi ces dernières servent. Quant aux différents circuits périphériques du ST et leurs rôles respectifs, ils sont répertoriés sur le schéma 15.3.

Pour chaque périphérique, les positions "mémoire" occupées (paires ou impaires) et les adresses sont précisées sur le schéma 15.4. On peut remarquer que pour la majorité des circuits l'instruction "MOVEP" s'avère utile, bien que pour d'autres elle ne s'impose pas. C'est par exemple le cas pour le circuit sonore ou le contrôleur vidéo dont certains registres sont en voie paire, d'autres en voie impaire.

Pour illustrer ces questions d'adressage de périphériques et pour agrémenter vos vacances studieuses, nous vous proposons un exemple de programmation du circuit sonore du ST : le YM-2149 (ou son équivalent).
Il y a, en assembleur, deux manières de programmer ce circuit : la première consiste à accéder directement aux registres du circuit grâce aux adresses précisées sur le schéma 15.4. Cependant, il va falloir assurer l'exécution de votre programme en mode superviseur sous peine de récupérer une exécrable erreur de bus (deux bombes, deux!) car le circuit se situe dans la zone périphérique "système".

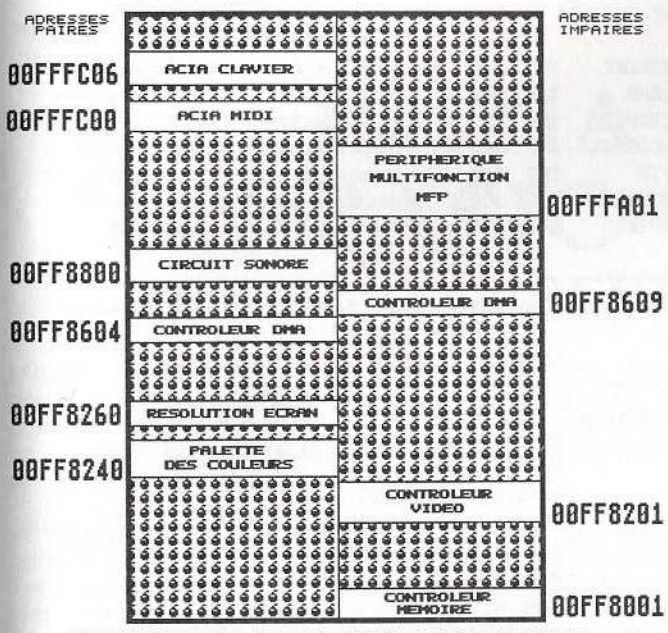


SCHEMA 15.2 EXEMPLES DE TRANSFERTS "MOVEP"



SCHEMA 15.3

LES CIRCUITS PERIPHERIQUES DE L'ATARI



ORGANISATION DE LA ZONE PERIPHERIQUE

SCHEMA 15.4

Par ailleurs, étant donné que certains de ces registres sont aussi utilisés pour les lecteurs de disques et la liaison série, nous allons nous intéresser sans plus tarder à la deuxième méthode, qui consiste à utiliser les fonctionnalités de la partie XBIOS du système d'exploitation...

Le but de l'exercice est de transformer votre pavé numérique en pavé musical (dans la mare...) sur lequel vous pourrez choisir, outre les notes, diverses enveloppes pour votre son. La correspondance entre les touches et les notes/enveloppes vous est indiquée ci-dessous:

Touche	
0	Note DO
1	Note RE
2	Note MI
3	Note FA
4	Note SOL
5	Note LA
6	Note SI
7	Enveloppe type "jeu video"
8	Enveloppe type "belle mère"
9	Enveloppe type "vaisseau spatial"
*	Réinitialisation du canal utilisé (son aigu)
Enter	Fin du programme (Ouf!)

Les autres touches ne sont pas utilisées, à chaque appui sur l'une d'entre elles, nous vous conseillons de réinitialiser le canal (touche '*').

La fonction XBIOS à utiliser est DOSOUND, pour laquelle il convient de placer l'adresse d'une chaîne de commande du circuit son sur la pile. L'explication des différentes chaînes de commandes n'est pas ici notre but, ceci dit nous vous en dirons l'essentiel pour pouvoir faire vos essais. Sachez néanmoins que beaucoup d'ouvrages sur le ST traitent du fonctionnement du circuit son...

Il nous faut tout d'abord une chaîne d'initialisation du canal A du circuit (le seul que nous utiliserons sur les trois canaux). Cette dernière est repérée par l'étiquette **init** dans le listing. D'autres chaînes permettent de faire jouer une note ou de programmer une enveloppe. Les équivalences (que vous ne manquerez pas de changer pour voir...) sont ci-dessus.

```
SI      EQU     150
LA      EQU     200
SOL     EQU     5
FA      EQU     70
MI      EQU     105
RE      EQU     180
DO      EQU     0

NO_WAVE EQU     15      ; pas d'enveloppe
WAVE1   EQU     8       ; première forme d'enveloppe
WAVE2   EQU     10          "
WAVE3   EQU     12          "
```

Pour les enveloppes, les valeurs de 0 à 15 sont possibles... La zone de données contient les chaînes de commandes du circuit que vous pouvez modifier par l'intermédiaire des équivalences de la page suivante.

```
        SECTION   DATA

table   DC.L      key_0,key_1,key_2,key_3,key_4,
        DC.L      key_5,key_6,key_7,key_8,key_9

init    DC.B      8,$F,7,$FE,FIN,COMMANDE
key_0   DC.B      1,3,0,DO,FIN,COMMANDE
key_1   DC.B      1,2,0,RE,FIN,COMMANDE
key_2   DC.B      1,2,0,MI,FIN,COMMANDE
key_3   DC.B      1,2,0,FA,FIN,COMMANDE
key_4   DC.B      1,2,0,SOL,FIN,COMMANDE
key_5   DC.B      1,1,0,LA,FIN,COMMANDE
key_6   DC.B      1,1,0,SI,FIN,COMMANDE
key_7   DC.B      8,$10,12,10,11,0,13,WAVE1,FIN,COMMANDE
key_8   DC.B      8,$10,12,10,11,0,13,WAVE2,FIN,COMMANDE
key_9   DC.B      8,$10,12,10,11,0,13,WAVE3,FIN,COMMANDE
key_star DC.B     8,NO_WAVE,FIN,COMMANDE
```

Dans la zone de données initialisées, l'étiquette "table" définit un tableau d'adresses des différentes chaînes de commande (key_x). Pour passer cette chaîne de commande à la fonction DOSOUND, nous utiliserons le mode d'adressage DEP8(An,Dn.w) avec An contenant l'adresse "table" et D0.W le déplacement par rapport à cette adresse. Ce déplacement est obtenu d'après le code ASCII de la touche frappée moins $30 (afin d'obtenir une valeur comprise entre 0 et 9), puis il y a une multiplication par 4 pour pointer sur l'adresse voulue, vu qu'une adresse est codée sur quatre octets... (voir le listing).

Nous vous proposons un pseudo-code, autour duquel toutes les fantaisies sont permises:

```
DEBUT
/* Startup */
/* Empiler la chaine d'initialisation */
/* Appel à DOSOUND */

REPETER
 /* Saisie clavier sans écho (CRAWCIN) */
 SI D1.B est NUL
  ALORS
   SI D0.B différent de '*'
    ALORS
     /* retrancher $30 de D0 */
      SI D0 non négatif
       ALORS
        SI D0 inférieur à 9
         ALORS
          /* Multiplication de D0 par 4 */
          /* init A0 avec l'adresse "table" */
          /* init A1 avec l'adresse de la chaine de cmde */
          /* empilement de A1 */
          /* appel à DOSOUND */
        FINSI
      FINSI
   SINON
    /* empiler l'adresse de la chaine de réinit. */
    /* appel à DOSOUND */
   FINSI
  SINON
   /* retour startup */
  FINSI
 JUSQU'A l'infini
FIN
```

Dans le Numéro 41, nous avions écrit un sous-programme effectuant une attente jusqu'à appui sur la touche 'ESC'. Nous allons ici réaliser pratiquement la même chose sauf que la touche à détecter est "ENTER". Le sous-programme retourne dans D0.B la valeur ASCII de la touche frappée et $FF dans D1 s'il s'agit de ENTER. Nous conseillons de reprendre en main votre collection de ST Mag pour bien assimiler cette partie du programme :

```
Hit_kbd  EQU      *
         MOVE.W   #CRAWCIN,-(A7)     Code de fonction
         TRAP     #1                 Appel GEMDOS
         ADDQ.L   #2,A7              Réalignement de la pile
         CMPI.B   #ENTER,D0          Test touche frappée
         SEQ      D1                 Si ENTER alors D1 = $FF
         RTS
```

Voilà, c'est tout! A vous de jouer. N'oubliez pas de faire l'édition de liens de votre programme avec le STARTUP.BIN ou d'insérer le STARTUP.S en source avant de générer votre ".PRG".
Outre les équivalences et la zone de données initialisée, le listing complet du programme (y compris le STARTUP pour les réfractaires de l'édition de liens) est le suivant :

```
SHRINK    EQU  $4A    ; code fonction, rend la mémoire
TERM      EQU  $4C    ; code fonction, Fin de programme
CRAWCIN   EQU  $07    ; code fonction, saisie clavier
DOSOUND   EQU  $20    ; code fonction, circuit sonore
FIN       EQU  $FF    ; premier octet de fin de commande
COMMANDE  EQU  $00    ; deuxième octet de fin de commande
ENTER     EQU  $0D    ; touche de fin de programme

*******
* Rajouter les équivalences des notes et enveloppes...

          SECTION  TEXT
*******
* Startup
          XREF     Main
          MOVE.L   4(A7),A0
          MOVE.L   $C(A0),D0
          ADD.L    $14(A0),D0
          ADD.L    $1C(A0),D0
          ADD.L    #$800,D0
          ANDI.L   #-2,D0
          LEA.L    0(A0,D0),A7
          MOVE.L   D0,-(A7)
          MOVE.L   A0,-(A7)
          CLR.W    -(A7)
          MOVE.W   #SHRINK,-(A7)
          TRAP     #1
          ADD.L    #12,A7
          TST.L    D0
          BNE      Out_gem
          JSR      Main
Out_gem
          MOVE.W   D0,-(A7)
          MOVE.W   #TERM,-(A7)
          TRAP     #1

*******
* Programme principal

Main      EQU      *
          PEA      init(PC)        ; initialise le canal A
          MOVE.W   #DOSOUND,-(A7)
          TRAP     #14
          ADDQ.L   #6,A7
```

```
Loop    BSR     Hit_key
        TST.B   D1              ; fin ?
        BNE     Retour
        CMPI.B  #"*",D0         ; reset du canal A
        BEQ     Not_wave
        SUBI.B  #$30,D0         ; touches 0 à 9 ?
        BMI     Loop
        CMPI.B  #9,D0
        BHI     Loop
        MULU    #4,D0           ; calcul offset table
        LEA     table(PC),A0
        MOVE.L  0(A0,D0.W),A1
        MOVE.L  A1,-(A7)
        MOVE.W  #DOSOUND,-(A7)
        TRAP    #14             ; play it...
        ADDQ.L  #6,A7
        BRA     Loop

Not_wave PEA    key_star(PC)
        MOVE.W  #DOSOUND,-(a7)
        TRAP    #14             ; programmation enveloppe
        ADDQ.L  #6,A7
        BRA     Loop
Retour  RTS

*******
Hit_key ...     A compléter...

*******

        SECTION DATA
        ...     A compléter...

        END
```

Noter que ce programme source et l'exécutable associé sont sur la "disquette magazine" du présent numéro (fichiers PLAY.S et PLAY.TOS). Bonne mumuse!
Il y a fort à parier que vous allez irriter les oreilles de votre petit frère avec tous les bruits bizarres que vos programmes inestimables distilleront tout au long de ces mois de vacances! Mettez-lui des "boules quiès" que vous enlèverez en septembre pour lui expliquer le principe des exceptions et des interruptions que vous aurez fièrement appris d'ici-là...

C. PASCALADA

INITIATION AU BASIC GFA - V

(SERIE N° 2)

L'heure est venue d'éclairer nos pas. Quel est donc notre objectif final, notre destinée secrète, notre voeu de connaissance ? Connaître la structure d'un micro ordinateur ! Et pour ce faire, il FAUT que vous maîtrisiez parfaitement les notions qui vont suivre.

Si vous souhaitez percer tous les mystères du codage binaire, vous devrez connaître les puissances de deux. Tous ceux qui haïssent viscéralement les mathématiques viennent de quitter bruyamment le rang. Revenez ! En fait, cette méthode de calcul est débilement simple, et vous ne pouvez que réussir à la pratiquer. Nous avons créé - à votre intention - plusieurs petits programmes qui vont vous y aider.

Les puissances de deux

Élever un nombre à la puissance N, consiste à multiplier ce nombre par lui-même, N fois de suite. Par exemple, 3 à la puissance 2 est égal à 9 ; 2 à la puissance 3 est égal à 8, etc. Pour ce qui nous concerne, seules les puissances du chiffre 2 présentent de l'intérêt. Ceci tient au fait qu'un bit a deux états, et que combiner plusieurs bits revient à combiner tous leurs états possibles.

Notation des puissances : comme vous le savez, l'élévation à la puissance est notée, en Basic, avec l'accent circonflexe. Par exemple, 2 à la puissance 3 sera noté :

```
x%=2^3
```

Table des puissances de deux : voici un petit programme qui sert à établir la table des puissances de 2.

```
'    TABLE_2.GFA
'
PRINT "Puissance","2 à la puissance"
PRINT
FOR puissance%=0 TO 20
  PRINT puissance%,2^puissance%
NEXT puissance%
```

La table des puissances de 2 est éditée ci-dessous. Elle commence à la puissance 0 et se termine (arbitrairement) à la puissance 20 :

Puissance	2 à la puissance
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576

Établir une table des puissances de 2 est d'une simplicité désarmante, puisqu'il suffit de commencer par 1 et de multiplier à chaque fois le résultat par 2.

ATTENTION : en général, on oublie que 2 à la puissance 0 est égal à 1 : tout chiffre élevé à la puissance 0 est égal à 1. De même, il faut se souvenir que 2 à la puissance 1 est égal à 2 : N à la puissance 1 est égal à N.

Le kilooctet : cette table nous offre l'occasion d'une digression notoirement importante. Les puissances de 2 s'infiltrent partout en informatique. Prenons-en un exemple simple et concret : la notation d'un kilooctet. Comme vous le savez, le préfixe "kilo" signifie ordinairement "mille" (un kilogramme = 1000 grammes ; un kilomètre = 1000 mètres, etc.). Vous êtes benoîtement prêt à en déduire qu'un kilooctet comporte 1000 octets...

EH BIEN NON. Un kilooctet, *par convention*, c'est 2 à la puissance 10, c'est-à-dire, si vous consultez notre table, 1024 octets ! La différence ne porte guère à conséquence, mais il est bon de savoir qu'un kilo "pèse plus lourd" en informatique.

Notation d'un mégaoctet : un méga correspond à un million. Ordinairement. Vous avez deviné qu'il n'en est pas de même en informatique, puisque un méga correspond à 2 à la puissance 20, soit 1.048.576 octets.

Vous comprenez pourquoi le ST 1040 a, en fait, 1.048.576 octets ; et le 520 en comporte 2 à la puissance 19, soit 524 288 octets ?

Test des puissances de deux : pour vous permettre de vérifier vos connaissances, nous vous proposons un petit programme d'auto-évaluation. Il effectue le tirage au sort d'une puissance comprise entre 0 et 7, et vous demande d'appliquer cette puissance au chiffre 2. Par exemple, il vous propose la puissance 3, et vous demande combien font 2 à la puissance 3 (réponse : 8) :

```
'       PUISS_2.LST
'       Test des puissances de deux
PRINT
PRINT "* Test des puissances de deux *"
puissance%=RANDOM(8)          ! Au hasard, Balthazar
REPEAT
  encore%=TRUE
  PRINT
  PRINT CHR$(27)+"p"
  PRINT "2 à la puissance ";puissance%
  PRINT CHR$(27)+"q"
  INPUT reponse%
  IF reponse%=2^puissance%
    PRINT
    PRINT "Bravo !"
  ELSE
    ALERT 1,"Bah, voyons...",1,"Refaire|Réponse",ret%
    IF ret%=1
      encore%=FALSE
    ENDIF
  ENDIF
UNTIL encore%
PRINT
PRINT "**** REPONSE ****"
PRINT
PRINT "2 à la puissance ";puissance%;" = ";2^puissance%
```

Conversion du binaire au décimal

Nous allons continuer de nous intéresser aux octets, et tenter de comprendre comment un octet binaire peut être directement converti dans le nombre décimal qui lui correspond. En clair, je vous donne un octet écrit en binaire, et vous me donnez sa valeur numérique.

Principe de conversion d'un octet : il y a quelques principes, qui vous mèneront infailliblement au résultat :
- numéroter les colonnes de la droite vers la gauche, en commençant par zéro (ceci vous donne le numéro du bit, de 0 à 7);
- mettre *au-dessus* de chaque colonne la puissance de deux correspondant au numéro du bit (de 1 à 128) ;
- multiplier la valeur du bit (0 ou 1) par la puissance de deux notée au-dessus de la colonne ;
- additionner chaque valeur obtenue.

Voici un schéma récapitulant ce que nous venons d'énoncer :

```
  128 64 32 16 8  4  2  1 <- 2 à la puissance numéro-du-bit
   7  6  5  4  3  2  1  0 <- numéro du bit

   1  0  0  1  0  1  1  1 <- octet

  Résultat :     128 * 1 = 128
             +    64 * 0 = 0
             +    32 * 0 = 0
             +    16 * 1 = 16
             +     8 * 0 = 0
             +     4 * 1 = 4
             +     2 * 1 = 2
             +     1 * 1 = 1

  Somme = 128 + 16 + 4 + 2 + 1 = 151
```

Avouez qu'il n'y a pas de quoi pousser mémère ! Inutile de préciser que nous avons noté les multiplications par zéro, dans cet exemple, mais qu'il est parfaitement superflu de le faire (dorénavant).

Comment programmeriez-vous une telle conversion du binaire au décimal ? Voici notre réponse, sous forme d'un petit programme :

```
'       BIN_DEC.GFA
'       Test de codage (binaire -> décimal)
octet%=RANDOM(256)
chn$=""                  ! Variable chaîne globale
PRINT
PRINT "* Test de conversion d'une valeur *"
PRINT "*    binaire en valeur décimale    *"
enonce
REPEAT
  encore%=TRUE
  INPUT "Valeur décimale équivalente ";reponse%
  IF reponse%=octet%
    PRINT
    PRINT "Bravo !"
  ELSE
    ALERT 1,"Bah, voyons...",1,"Refaire|Réponse",ret%
    IF ret%=1
      encore%=FALSE
    ENDIF
  ENDIF
UNTIL encore%
PRINT
PRINT "**** REPONSE ****"
corrige
END
```

```
PROCEDURE enonce
  '        --- Globales : chn$, octet%
  PRINT CHR$(27)+"p"
  PRINT "  7  6  5  4  3  2  1  0 <- bit  "
  chn$=STRING$(8,"0")+BIN$(octet%)
  edit_bin
  PRINT CHR$(27)+"q"
RETURN
'
PROCEDURE corrige
  LOCAL i%,copie_octet%
  '     --- Globale : octet%
  PRINT CHR$(27)+"p"
  PRINT " 128 64 32 16 8  4  2  1 <- 2^bit"
  PRINT CHR$(27)+"q"
  edit_bin
  PRINT
  copie_octet%=octet%
  FOR i%=7 DOWNTO 0
    IF octet%>=2^i%
      PRINT " + ";2^i%;
      SUB octet%,2^i%
    ENDIF
  NEXT i%
  PRINT " = ";copie_octet%
  ~BIOS(2,2)
RETURN
'
PROCEDURE edit_bin
  LOCAL i%
  '     --- Globale : chn$
  chn$=RIGHT$(chn$,8)
  FOR i%=1 TO 8
    PRINT "  ";MID$(chn$,i%,1);
  NEXT i%
  PRINT
RETURN
```

Ce programme a pour effet d'afficher sur votre écran un résultat qui ressemble à ceci :

```
* Test de conversion d'une valeur *
*    binaire en valeur décimale   *

  7  6  5  4  3  2  1  0 <- bit
  1  0  0  1  0  1  1  1

Valeur décimale équivalente ?

151

Bravo !

**** REPONSE ****

 128 64 32 16 8  4  2  1 <- 2^bit

  1  0  0  1  0  1  1  1

 + 128 + 16 + 4 + 2 + 1 = 151
```

De fait, vous verrez quelques différences sur votre écran, puisque notre programme fait appel à des *séquences Escape*, qui sont à l'origine de l'écriture de caractères en inverse vidéo (blanc sur fond noir). Les séquences Escape apparaissent comme ceci :

```
PRINT CHR$(27)+"p"
```

(ne vous préoccupez pas pour l'instant, de savoir ce que cette expression peut signifier, l'explication vous sera donnée - pour un prix modique - à la fin de cet article.)

Revenons à notre programme. Nous avons pris arbitrairement la valeur 151, mais il est bien évident que la valeur qui vous sera demandée sera différente (tirage au sort grâce à RANDOM).

ATTENTION : rappelez-vous toujours que la numérotation des bits se fait de droite à gauche, c'est-à-dire dans le sens contraire du sens habituel ; la première valeur (colonne de droite) porte le numéro 1.

Edition décimale d'une valeur binaire : le programme qui va suivre effectue la saisie d'un octet, écrit en binaire (par l'utilisateur), et le convertit en décimal. Vous devriez être capable de faire vous-même cette conversion, et de regarder ensuite notre solution. Mais attention, il faut effectuer une *saisie binaire* composée uniquement de 0 et de 1 !

```
'      ED_DEC.GFA
'      Edition décimale d'une valeur binaire
PRINT
PRINT "Valeur binaire à éditer en décimal ?"
PRINT "*       8 bits seulement !         *"
PRINT
chn$=""              ! Variable chaîne de caractères globale
PRINT CHR$(27)+"p"
saisie_bin(8)
PRINT CHR$(27)+"q"
longueur%=LEN(chn$)
val%=0
j%=1
FOR i%=longueur%-1 DOWNTO 0
  IF MID$(chn$,j%,1)="1"
    val%=val%+(2^i%)
  ENDIF
  INC j%
NEXT i%
PRINT
PRINT "Valeur décimale = ";val%
'
PROCEDURE saisie_bin(n_car%)      ! Saisie contrôlée
  LOCAL tch%,car$,i%,ret%
  '     --- Globale : chn$
  i%=1
  REPEAT
    tch%=BIOS(2,2)
    car$=CHR$(tch%)
    IF ASC(car$)=13                ! Touche RETURN
      i%=n_car%+1
    ELSE IF car$="0" OR car$="1"
      PRINT car$;
      chn$=chn$+car$
      INC i%
    ELSE
      ALERT 3,"Des 0 et des 1 SVP !",1,"Ok...",ret%
    ENDIF
  UNTIL i%>n_car%
  PRINT
RETURN
```

Pour sortir de la saisie binaire, frappez la touche RETURN ou ENTER (elles ont le même code ASCII : 13).

Le coeur de ce programme, la conversion de la valeur binaire en valeur décimale, est plus facile à programmer que la saisie binaire elle-même. Si vous ne maîtrisez pas totalement cette routine de saisie, n'hésitez pas à y revenir ultérieurement, elle n'est pas au centre du débat.

Ecrire la prochaine valeur binaire

Votre formation à l'écriture binaire va progresser d'une étape. Notre objectif est simple : savez-vous écrire les valeurs binaires dans un *ordre croissant* ? Voici, pour cadrer le problème, les cinq premières valeurs croissantes (pour simplifier, nous avons supprimé les zéros inutiles, situés à gauche):

```
  0
  1
 10
 11
100
```

Pour vous aider, nous allons vous proposer une méthode très simple à retenir. *Elle consiste à lire mentalement la suite de chiffres, comme s'il s'agissait de valeurs décimales (et non binaires).* Par exemple, la liste précédente se lirait : "zéro, un, dix, onze, cent,...".

Pour écrire la liste croissante, posez-vous cette question : quel est le chiffre immédiatement supérieur au précédent, mais s'écrivant uniquement avec des 0 et des 1 ? Nous pourrions poursuivre la liste en disant : "cent un, cent dix, cent onze, mille, mille un, etc.".

```
  101
  110
  111
 1000
 1001
```

Vous comprenez ce simplissime principe ? Voici l'occasion de vous tester grâce à un programme de notre cru :

```
DEFINT "a,z"   ! Toutes variables entières
PRINT
PRINT "***** Test de codage binaire *****"
PRINT
PRINT "* Donnez la valeur immédiatement *"
PRINT "* supérieure à celle présentée   *"
PRINT "* en utilisant uniquement        *"
PRINT "* des 0 et des 1 !               *"
PRINT
mini=1    ! Valeurs modifiables
maxo=100  ! Ne pas dépasser 256
PRINT "Valeur initiale = ";maxo
PRINT
val=mini
REPEAT
  PRINT "Valeur immédiatement supérieure"
  PRINT "à (décimal = ";val;") :"
  PRINT
  PRINT CHR$(27)+"p";BIN$(val);CHR$(27)+"q"
  PRINT " ?"
  chn$=""
  encore%=FALSE
  saisie_bin(8)
  IF @conv_bin_dec=val+1
    PRINT
    PRINT "** Bravo ! On continue... **"
    PRINT
    INC val
  ELSE
    ALERT 1,"Bah, voyons...",1,"Refaire|Réponse",ret%
    IF ret%=2
      encore%=TRUE
    ENDIF
  ENDIF
UNTIL val=maxo OR encore%
PRINT
PRINT "*** REPONSE ***"
PRINT
PRINT "Réponse correcte = ";BIN$(val+1)
END
'
FUNCTION conv_bin_dec
  LOCAL longueur%,valeur%,j%,i%
  '     --- Globale : chn$
  longueur%=LEN(chn$)
  valeur%=0
  j%=1
  FOR i%=longueur%-1 DOWNTO 0
    IF MID$(chn$,j%,1)="1"
      ADD valeur%,2^i%
    ENDIF
    INC j%
  NEXT i%
  RETURN valeur%
ENDFUNC
```

N'oubliez surtout pas d'insérer la procédure 'saisie_bin', abordée précédemment. Vous pouvez modifier à votre guise les valeurs de 'mini' et de 'maxo' qui sont les valeurs de début et de fin. Si vous utilisez GfA 2, supprimez la ligne d'instruction DEFINT, et placez le suffixe '%' à la fin de chaque identificateur de variable entière.

Retour sur les types

Nous allons procéder à une récapitulation des types de variable disponibles en GfA 3. Essayez de ne pas tout embrouiller, ce n'est pas compliqué, mais on peut perdre pied !

Octet, mot court, mot long : les octets n'ont déjà presque plus de secrets pour vous. L'heure est venue de faire connaissance avec d'autres types de données : les mots courts et les mots longs.

Le terme "mot court" est la traduction de "short word" ou plus simplement de "word" ; "mot long" est la traduction de "long word", mais vous rencontrerez souvent le terme "integer" pour désigner un mot long (ce terme est la contraction de "long integer" qui signifie "entier long"). Puisque nous sommes dans les anglicismes, rappelons que "octet" se dit "byte".

Par convention, quand on parle d'un "mot", il est question d'un *mot court*. Un mot est composé de 16 bits. On peut donc considérer qu'il est assimilable à *deux octets*. Voici un schéma qui en donne une représentation cohérente avec nos descriptions précédentes :

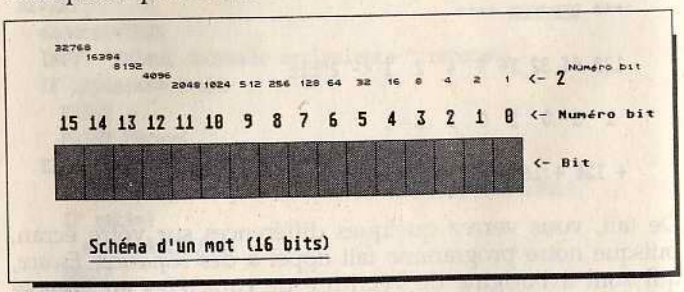


Schéma d'un mot (16 bits)

Tous les principes de calcul et de notation binaire, rencontrés jusqu'à maintenant, sont pleinement applicables. Vous vous souvenez que la valeur entière maximale, que l'on puisse représenter sur un octet, est égale à 255 ? (Mais si, rappelez-vous : un octet est composé de huit bits, la valeur maximale c'est donc : 128+64+32+16+8+4+2+1 = 255.)

En suivant le même principe, la valeur maximale, représentable sur un mot, est égale à 65535.

Calcul de la valeur maximale théorique d'un mot court : rappelez-vous de la méthode employée pour calculer la valeur maximale d'un octet. Elle s'applique à un mot, lequel fait appel à 16 bits, et donc à 16 valeurs (somme des 16 premières puissances de 2) :

32768+16384+8192+4096+2048+1024+512+256+128+64+32 +16+8+4+2+1 = 65535

Un entier est une quantité signée : il est pratique de disposer de quantités signées. Ceci permet de représenter des quantités négatives, nulles ou positives, en fonction des cas.

Or, la norme, créée par GfA, fait qu'un *octet* permet de représenter uniquement des valeurs entières, comprises entre 0 et 255. En revanche, il a été décidé de donner un signe aux entiers courts (word) ou longs (integer).

Par convention, pour représenter le signe, on utilise le bit situé à gauche (bit numéro 15).

C'est ainsi qu'un entier court (word) connaît des valeurs comprises entre -32768 et +32767. Vous pouvez donc représenter tout nombre entier, compris entre ces deux valeurs, y compris la valeur zéro.

ATTENTION : par conséquent, un mot (word) n'est PAS compris entre 0 et 65535, comme on pouvait s'y attendre, mais entre -32768 et +32767 !

Remarque : si vous additionnez le nombre *d'états* possibles, en incluant la valeur zéro, vous obtenez bien : 32768+32767+1 = 65536 (soit 2 à la puissance 16).

Les mots longs (integer) : ils sont composés de 32 bits, soit quatre octets, et permettent de représenter des quantités signées. L'intervalle des valeurs possibles est compris entre -2147483648 et +2147483647. Le suffixe employé est '%'.

Tableau récapitulatif : voici un tableau récapitulatif, qui doit vous prouver que tout n'est pas si compliqué. L'important, c'est que vous vous souveniez dans quels cas utiliser un entier plutôt qu'un autre :

Type	anglais	suffixe	taille mémoire	mini	maxi
octet	byte	|	8 bits (1 octet)	0	255
mot	word	&	16 bits (2 octets)	-32768	+32767
mot long	integer	%	32 bits (4 octets)	-2147483648	+2147483647

Remarque : d'un point de vue théorique, rien n'empêche de représenter des quantités signées sur un octet. Nous pourrions donc y représenter des valeurs comprises entre -128 et +127 (le bit numéro 7 serait utilisé comme "bit de signe"), mais la convention utilisée par GfA n'utilise tout simplement PAS cette convention ! En GfA, un octet représente un entier compris entre 0 et 255. C'est comme ça.

Les séquences escape

Dans ce qui a précédé nous avons, sans vergogne, employé des séquences escape. Que recouvre ce terme aux connotations ouvertement anglo-saxonnes ? Voici une séquence escape :

```
PRINT CHR$(27)+"p"
```

La touche ESC, en haut et à gauche de votre clavier, est la touche ESCAPE, le code ASCII qui lui correspond est le code '27'. Une séquence escape, c'est donc l'envoi du caractère escape, suivi d'un ou plusieurs caractères. Pour transmettre le caractère escape, il suffit d'employer CHR$(27), fonction qui renvoie un caractère (celui correspondant au code ASCII transmis).

Soit, mais le second caractère de la séquence ? Ce second caractère dépend de conventions diverses et variées. Sur le ST, les lettres suivantes ont les effets suivants (c'est selon) :

- A curseur vers le haut
- B curseur vers le bas
- C curseur vers la droite
- D curseur vers la gauche
- e activation curseur
- E efface l'écran
- f désactivation curseur
- H curseur Home
- p inverse vidéo (lettres blanches, fond noir)
- q vidéo normale (lettres noires, fond blanc)

Vous avez reconnu les deux dernières séquences de cette liste (non exhaustive), puisqu'elles ont été employées dans nos programmes précédents.

Dernière remarque avant la fin de nos émissions : ces séquences escape sont des ordres, donnés à ce qu'il est convenu d'appeler l'émulateur VT 52, qui n'est autre qu'un éditeur intégré. Cet éditeur permet de manipuler des textes, mais il se marie mal avec le graphisme. À employer avec circonspection !

Christophe Castro

DANS LE COURANT DE L'ETE, ST MAGAZINE AURA SON TELECHARGEMENT!

RETROUVEZ LES SOFTS PRESSIMAGE ET FREEWARE, EN VOUS PROCURANT LE SOFT DE TELECHARGEMENT A LA BOUTIQUE DE PRESSIMAGE. NOUS LE DIFFUSONS POUR LA MODIQUE SOMME DE 15 FRANCS PORT COMPRIS. IL SERA DISPONIBLE D'ICI PEU.

3615 STMAG

STOS PRATIQUE (8)

Tiens? Revoilà la rubrique STOS! Elle se faisait rare ces derniers mois. Quelques petits problèmes militaires m'ont en effet empêché d'utiliser mon ordinateur... (merci mon colonel!)

STOS TRUCS/STOS TRUCS/STOS TRUCS/STOS TRUCS

Les nouvelles ROM 1.62
Atari vient encore de frapper avec de nouvelles nouvelles ROM! Bien entendu le STOS ne fonctionne plus (et, je pense, ne sera pas le seul !). Mais pas de panique, je suis en train d'écrire un petit programme domaine public qui fera l'adaptation. Vous pourrez très bientôt le télécharger sur le 3615 STMAG. Bon c'est pas tout ça, mais il faut que je trouve quelqu'un qui possède ces fameuses nouvelles-nouvelles Roms pour pouvoir faire l'adaptation!

A propos des versions.
La dernière version du STOS est la version 2.5. Le programme de chargement (BASIC205.PRG) affiche juste avant de passer au basic la dernière version de la ROM avec laquelle le STOS fonctionne. C'est-à-dire TOS 1.6 jusqu'à ce jour. Cela ne veut pas dire que le STOS ne fonctionne qu'avec le TOS 1.6, mais avec toutes les versions JUSQU'AU 1.6!

DEMOS DRAW
Voici trois petites démonstrations utilisant DRAW et une rotation de couleurs, écrites par Peter J. Hickmann, un confrère anglais. Vive l'Europe! Amusez-vous à modifier les constantes de dessin, les résultats sont assez étonnants!

```
10 rem --------------------------------
11 rem Demo draw 1
12 rem Par P.J.Hickman
13 rem --------------------------------
20 key off : mode 0 : curs off : flash off : hide on
30 rem Cycle les couleurs
40 A$="(000,1)(001,1)(002,1)(003,1)(004,1)(005,1)(006,1)
   (007,1)(006,1)(005,1)(004,1)(003,1)(002,1)(001,1)"
50 for N=1 to 15 : flash N,A$ : wait vbl : next
60  :
70 rem Definition des variables
80 TW#=2*pi
90 XCENTER=151
100 YCENTER=93
110 R1=150
120 R2=35
130 R3=65
140 R4=85
150 I#=pi/512
160 SET#=pi/3
170 C=1
180  :
190 rem Boucle principale
200 for T#=0 to TW# step I#
210 X1#=R1*cos(T#)+XCENTER
220 Y1#=R2*sin(T#)+YCENTER
230 X2#=R3*cos(T#+SET#/0.2)+XCENTER
240 Y2#=R4*sin(T#)+YCENTER
250 ink C
260 draw X1#,Y1# to X2#,Y2#
270 C=C mod 15+1
280 next T#
290  :
300 wait key : default
```

```
10 rem --------------------------------
11 rem Demo draw 2
12 rem Par P.J.Hickman
13 rem --------------------------------
20 key off : mode 0 : curs off : flash off : hide on
30 rem Cycle les couleurs
40 A$="(000,1)(100,1)(200,1)(300,1)(400,1)(500,1)(600,1)
   (700,1)(600,1)(500,1)(400,1)(300,1)(200,1)(100,1)"
50 for N=1 to 15 : flash N,A$ : wait vbl : next
60  :
70 rem Definition des variables
80 TW#=2*pi
90 XCENTER=151
100 YCENTER=93
110 R1=115
120 R2=85
130 R3=40
140 R4=45
150 I#=pi/8 : I2#=pi/20
160 i2#=pi/20
170 C=1
180  :
190 rem Boucle principale
200 for T#=0 to TW# step I#
205 for T2#=0 to TW# step I2#
210 X1#=R1*cos(T2#)+XCENTER
220 Y1#=R2*sin(T2#)+YCENTER
230 X2#=R3*cos(T#)+XCENTER
240 Y2#=R4*sin(T#)+YCENTER
250 ink C
260 draw X1#,Y1# to X2#,Y2#
270 C=C mod 15+1
280 next T2# : next T#
290  :
300 wait key : default
```

```
10 rem --------------------------------
11 rem Demo draw 3
12 rem Par P.J.Hickman
13 rem --------------------------------
20 key off : mode 0 : curs off : flash off : hide on
30 rem Cycle les couleurs
40 A$="(000,1)(110,1)(220,1)(330,1)(440,1)(550,1)(660,1)
   (770,1)(660,1)(550,1)(440,1)(330,1)(220,1)(110,1)"
50 for N=1 to 15 : flash N,A$ : wait vbl : next
60  :
70 rem Definition des variables
80 TW#=2*pi
90 XCENTER=151
100 YCENTER=93
110 R1=100
120 R2=85
150 I#=pi/640
160 LOBES#=3
170 C=1
180  :
190 rem Boucle principale
200 for T#=0 to TW# step I#
205 X1#=R1*cos(T#*LOBES#)+XCENTER
210 Y1#=R2*sin(T#)+YCENTER
220 ink C
230 draw XCENTER,YCENTER to X1#,Y1#
240 C=C mod 15+1
250 next T#
290  :
300 wait key : default
```

Les codes de contrôle texte

Le STOS basic communique avec la routine de gestion des fenêtres (et du texte en général) par l'intermédiaire de codes de contrôle, ou si vous préférez de CHR$(). Ainsi, l'instruction HOME se contente d'envoyer le code 31 aux fenêtres...

- 1: un mot vers la gauche
- 2: un mot vers la droite
- 3: Curseur vers la gauche
- 4: Scrolling vers le haut
- 5: Scrolling vers le bas
- 6: Return (va à la fin de la ligne REELLE, en comptant les caractères insérés);
- 7: Delete
- 8: BackSpace
- 9: Curseur vers la droite
- 10: Curseur vers le bas
- 11: Curseur vers le haut
- 12: Clw
- 13: Retour chariot
- 14: Writing 1
- 15: Writing 2
- 16: Writing 3
- 17: Apparition du curseur
- 18: Arrêt de l'inverse vidéo
- 19: Arrêt des caractères ombrés
- 20: Arrêt du curseur
- 21: Mise en marche inverse vidéo
- 22: Mise en marche des caractères ombrés
- 23: Autorisation des scrollings
- 24: Positionne le curseur à la fin de la ligne
- 25: Scroll off
- 26: Delete
- 27: Echappement
- 28: Mise en marche de l'insertion
- 29: Arrêt du soulignement
- 30: HOME
- 31: Mise en marche du soulignement

ZENITH!

Dans la dernière rubrique, nous avons ajouté l'éditeur de terrain à ce fabuleux programme. Pour le rendre encore plus fabuleux, nous allons ce mois-ci ajouter un éditeur des blocs constituant le terrain : une sorte de mini Degas (mais alors vraiment mini!). Zénith! est un développement "en direct", en voici la preuve : il va vous falloir remplacer certaines parties du programme pour permettre l'insertion de l'éditeur de bloc. Commençons par la routine de lecture des datas des menus éditeur. Nous allons avoir besoin d'un autre menu, il faut donc ajouter une dimension au tableau de stockage des menus. Tapez :

```
Delete 600-650 <return>
```

Puis

```
600 dim M_X(2,15),M_Y(2,15),M_M$(2,15),M_GO(2,15),M_MAX(2)
605 restore 52000 : NM=0
610 inc NM : M_MAX(NM)=1
615 read T : if T>0 then M_X(NM,M_MAX(NM))=T :
    read M_Y(NM,M_MAX(NM)),M_M$(NM,M_MAX(NM)),
    M_GO(NM,M_MAX(NM)) : inc M_MAX(NM) : goto 615
620 dec M_MAX(NM)
625 if T=0 then 610
```

Il s'agit pratiquement de la même routine en deux dimensions. La fin d'un menu est marquée par un 0, la fin de tous les menus par un -1. Autre partie ayant subi des modifications, l'initialisation de l'éditeur de terrain :

```
Delete 4000-4190 <return>
```

Puis:

```
4000 hide on : auto back off : update off
4005 DTERRAIN=start(PUZZLE)+NBLX*NBLY*2 :
     FTERRAIN=start(PUZZLE)+NBLX*NBLINE*2
4010 ADJOUEUR=DTERRAIN : BL=0
4024 rem > Dessin du terrain
4025 logic=TRAVAIL
4030 screen copy FOND to logic
4035 ink 0 : bar 1,YHO to XGOCH-2,YBA-2 :
     bar XDROI+1,YHO to 318,YBA-2
4040 reset zone
4045 NM=1 : gosub 12050
4050 logic=back
4055 set zone 100,XGOCH,YHO to XDROI-1,YBA-1
4060 set zone 101,XDROI+16,YHO+22 to
     XDROI+16+32,YHO+22+16
4065 gosub 12400
4070 gosub 12100
4075 OLDZM=0
4100 repeat
4105 XM=x mouse : YM=y mouse : KM=mouse key
4110 sprite 1,XM,YM,3 : redraw : ZM=zone(1) :
     screen copy TRAVAIL to logic
4111 if ZM=101 and KM=1 then 7000
4115 if ZM=100 and KM<>2 then
     screen copy BOUTS,XB,YB,XB+16,YB+16 to
     logic,XM,((YM-YHO) and $FFF0)+YHO : ink 3 :
     set mark 6,32 : polymark XM,YM
4120 if ZM<>100 or KM=2 then ink 3 : set mark 2,32 :
     polymark XM,YM
4125 if ZM=100 and KM=2 then ink 3 : set mark 5,32 :
     polymark XM,YM
4130 logic=TRAVAIL
4135 if ZM=100 then XT=(XM-XGOCH)/16 :
     YT=(ADJOUEUR-DTERRAIN)/(NBLX*2)+(YBA-YM-1)/16 :
     locate 1,20 : print "X:";XT;" "; : locate 1,21 :
     print "Y:";YT;" ";
4140 if ZM<100 then locate 1,20 : print "     "; :
     locate 1,21 : print "     ";
4145 if ZM=OLDZM then 4160
4150 if OLDZM then I=0 : N=OLDZM : gosub 12000 : OLDZM=0
4155 if ZM and (ZM<100) then I=1 : N=ZM : gosub 12000 :
     OLDZM=ZM
4160 logic=back : screen swap : wait vbl
4165 until KM<>0 and ZM<>0
4170 if ZM<>100 then gosub M_GO(NM,ZM) : goto 4100
4175 gosub 4500 : goto 4100
```

Les changements par rapport à la version précédente sont :

- Regroupement de la partie "initialisation de l'éditeur" après les lignes 4025;
- Utilisation d'une routine de dessin du menu, en 12050, NM contenant le numéro du menu à afficher;
- Passage à deux dimensions des tableaux de menu;
- Ajout d'une nouvelle zone de test (numéro 101) au niveau du bloc courant. En cliquant sur ce bloc, on passera à l'éditeur de blocs;
- Un nouveau (et plus beau) pointeur lorsque l'on dessine le terrain...

Continuons les changements:

```
Delete 12000-12050
```

Puis:

```
12000 X1=xgraphic(M_X(NM,N))-4 : Y1=ygraphic(M_Y(NM,N))-4
12005 X2=X1+len(M_M$(NM,N))*8+7 : Y2=Y1+15
12010 ink 1 : if I then inverse on else inverse off
12015 box X1,Y1 to X2,Y2
12020 locate M_X(NM,N),M_Y(NM,N) : print M_M$(NM,N);
12025 inverse off : return
12047 rem-----------------------------------
12048 rem  Editeur: dessin du menu NM
12049 rem-----------------------------------
12050 I=0
12055 for N=1 to M_MAX(NM)
12060 gosub 12000
12065 set zone N,X1,Y1 to X2,Y2
12070 next N
12075 return
```

Rien de bien compliqué ici : le passage à deux dimensions de la routine 12000, et une nouvelle routine dessinant automatiquement le menu NM (12050)...

Voici maintenant le corps de l'éditeur de blocs :

```
6997 rem-----------------------------------
6998 rem   Editeur de blocs
6999 rem-----------------------------------
7000 CINK=1 : CPAT=20 : XXBL=XGOCH+16 : YYBL=YHO+16
7005 CLX=XDROI+16 : CLY=YHO+22
7010 MZM=5 : XZM=160-MZM*8 : YZM=YHO+16
7024 rem > Init de l'ecran
7025 logic=TRAVAIL : screen copy FOND to logic
7030 ink 0 : bar XGOCH,YHO to XDROI-1,YBA-1
7035 reset zone
7040 NM=2 : gosub 12050 : OLDZM=0
7045 ink 1 : set paint 2,10,1 :
     bar XZM-8,YZM-8 to XZM+MZM*16+8,YZM+MZM*16+8
7050 set paint 1,1,0
7055 for C=0 to 15 : X=XGOCH+C*12 : Y=YBA-16 : ink C :
     bar X,Y to X+12,YBA-1 :
     set zone 50+C,X,Y to X+11,YBA-1 : next 7060 gosub 7250
7065 set zone 100,XZM,YZM to XZM+MZM*16-1,YZM+MZM*16-1
7070 logic=back : screen copy TRAVAIL to back
7075 gosub 12400 : gosub 7300
7095 set zone 100,XZM,YZM to XZM+MZM*16-1,YZM+MZM*16-1
7100 repeat
7105 XM=x mouse : YM=y mouse : KM=mouse key
7110 sprite 1,XM,YM,3 : redraw : ZM=zone(1) :
     screen copy TRAVAIL to logic
7115 if ZM<100 then ink I : set mark 2,32 :
     polymark XM,YM
7120 if ZM=100 and KM<>2 then ink I : set mark 4,2 :
     polymark XM,YM
7125 if ZM=100 and KM=2 then ink I : set mark 3,2 :
     polymark XM,YM
7130 inc I : I=I mod 16
7140 logic=TRAVAIL
7145 if ZM=OLDZM then 7160
7150 if OLDZM then I=0 : N=OLDZM : gosub 12000 : OLDZM=0
7155 if ZM and (ZM<50) then I=1 : N=ZM : gosub 12000 :
     OLDZM=ZM
7160 logic=back : screen swap : wait vbl
7165 until KM<>0 and ZM<>0
7170 if ZM<50 then gosub M_GO(NM,ZM) : goto 7100
7175 if ZM=100 then 7200
7180 if ZM<99 then CINK=ZM-50 : gosub 7325 :
     goto 7100
7185 inc CPAT : if CPAT>24+12 then CPAT=1
7190 gosub 7325 : goto 7100
7200 XX=(XM-XZM)/MZM : YY=(YM-YZM)/MZM
7205 CI=CINK : if KM=2 then CI=0
7210 if ACT$<>"Plot" and KM=2 then gosub 7250 :
     while mouse key : wend : goto 7100
7215 gosub ACTION : goto 7100
7249 rem > Plot
7250 ACTION=7255 : ACT$="Plot" : gosub 7325 : return
7255 logic=TRAVAIL : ink CI : set paint 1,1,0 :
     bar XZM+XX*MZM,YZM+YY*MZM to
     XZM+(XX+1)*MZM-1,YZM+(YY+1)*MZM-1
7260 plot XDROI+16+XX,YHO+22+YY :
     plot XDROI+32+XX,YHO+22+YY
7265 logic=BOUTS : plot XB+XX,YB+YY
7270 logic=back : return
7299 rem > Zoom du bloc
7300 zoom TRAVAIL,CLX,CLY,CLX+16,CLY+16 to
     TRAVAIL,XZM,YZM,XZM+MZM*16-1,YZM+MZM*16-1
7305 clip 0,0 to 319,199 : return
7309 rem > Bloc precedent
7310 gosub 4350 : gosub 7300 : return
7314 rem > Bloc suivant
7315 gosub 4375 : gosub 7300 : return
7324 rem > Changement de couleur / Impression action
7325 if CPAT<=12 then CPAT1=2 : CPAT2=CPAT else
     CPAT1=2 : CPAT2=CPAT-12
7330 logic=TRAVAIL : ink CINK : set paint CPAT1,CPAT2,1 :
     bar 4,177 to XGOCH-4,196 :
     set zone 99,4,177 to XGOCH-4,196
7335 pen 1 : locate 2,23 : print ACT$;
7340 cls TRAVAIL,0,XXBL,YYBL to XXBL+16,YYBL+16 :
     if BL$<>"" then screen$(TRAVAIL,XXBL,YYBL)=BL$
7345 logic=back : return
7349 rem > Routine generale ACTION
7350 ACTION=7355 : ACT$=ACT1$ : gosub 7325 : return
7355 X1=XX : Y1=YY : while mouse key :
     wend 7360 ACTION=7365 :
     ACT$=ACT2$ : gosub
7325 : return
7365 logic=TRAVAIL : clip CLX,CLY to CLX+16,CLY+16
7370 ink CI : gosub ACT
7375 while mouse key : wend
7380 screen copy TRAVAIL,CLX,CLY,CLX+16,CLY+16 to
     BOUTS,XB,YB
7385 gosub 12400 : gosub 7300 : logic=back
7390 while mouse key : wend
7395 goto 7350
7399 rem > Lignes
7400 ACT1$="Li.1" : ACT2$="Li.2" : ACT=7405 : goto 7350
7405 draw CLX+X1,CLY+Y1 to CLX+XX,CLY+YY : return
7424 rem > Box
7425 ACT1$="Bo.1" : ACT2$="Bo.2" : ACT=7430 : goto 7350
7430 box CLX+X1,CLY+Y1 to CLX+XX,CLY+YY : return
7449 rem > Bar
7450 ACT1$="Ba.1" : ACT2$="Ba.2" : ACT=7455 : goto 7350
7455 set paint CPAT1,CPAT2,1 : bar CLX+X1,CLY+Y1 to
     CLX+XX,CLY+YY : return
7474 rem > Disc
7475 ACT1$="Di.1" : ACT2$="Di.2" : ACT=7480 : goto 7350
7480 DX=XX-X1 : DY=YY-Y1 : R=sqr(DX*DX+DY*DY)
7485 if R then circle CLX+X1,CLY+Y1,R
7490 return
7499 rem > Clr
7500 ACT1$="Clr?" : ACT2$="Ok!!" : ACT=7505 : goto 7350
7505 cls TRAVAIL,0,CLX,CLY to CLX+16,CLY+16 : return
7524 rem > Pousse le bloc
7525 BL$=screen$(TRAVAIL,CLX,CLY to CLX+16,CLY+16) :
     gosub 7325 : return
7549 rem > Copie le bloc
```

```
7550 if BL$<>"" then screen$(BOUTS,XB,YB)=BL$ :
     gosub 12400 : gosub 7300
7555 return
7574 rem > Voir
7575 gosub 12100 : screen copy TRAVAIL to physic
7580 while mouse key : wend : repeat : until mouse key
7585 pop : goto 7025
7599 rem > Editer
7600 pop : clip 0,0 to 319,199 : goto 4025
7649 rem > Symetrie gauche/droite
7650 logic=TRAVAIL
7655 for Y=0 to 15
7660 for X=0 to 15
7665 plot CLX+16-X,CLY+Y,point(CLX+16+X,CLY+Y)
7670 next X
7675 next Y
7680 screen copy TRAVAIL,CLX,CLY,CLX+16,CLY+16 to
     BOUTS,XB,YB
7685 gosub 12400 : gosub 7300 : logic=back
7690 return
7699 rem > Symetrie haut/bas
7700 for Y=0 to 15
7705 screen copy TRAVAIL,CLX+16,CLY+15-Y,CLX+32,CLY+16-Y
     to TRAVAIL,CLX,CLY+Y
7710 next Y
7715 goto 7680
```

Et enfin les nouveaux datas définissant le menu de l'éditeur de blocs :

```
52060 data 0
52100 data 1,YHO/8+1," LINE ",7400
52105 data 1,YHO/8+3," BOX  ",7425
52110 data 1,YHO/8+5," BAR  ",7450
52115 data 1,YHO/8+7," DISC ",7475
52120 data 1,YHO/8+9," CLR  ",7500
52125 data 1,YHO/8+11," STOC ",7525
52126 data 1,YHO/8+13," RECU ",7550
52130 data XDROI/8+1,YHO/8+1," ++++ ",7310
52135 data XDROI/8+1,YHO/8+7," ---- ",7315
52140 data XDROI/8+1,YHO/8+10,"G<==>D",7650
52145 data XDROI/8+1,YHO/8+12,"H<==>B",7700
52150 data XDROI/8+1,YHO/8+14," VOIR ",7575
52155 data XDROI/8+1,YHO/8+16," EDIT ",7600
52160 data-1
```

Utilisation:

Sous l'éditeur de terrain, cliquez sur le bloc sélectionné (en haut à droite de l'écran), et vous passez à l'édition du bloc proprement dit :
- La sélection du bloc édité se fait comme toujours en cliquant sur + et - . Notez le ralentissement dû au ZOOM;
- Vous dessinez avec le bouton gauche de la souris, et effacez avec le bouton droit;
- Vous choisissez la couleur courante en cliquant dans la palette. La couleur sélectionnée, le motif de remplissage des surfaces ainsi que le mode graphique en fonction sont affichés en bas à gauche de l'écran;
- Vous sélectionnez une fonction graphique en cliquant sur le bouton correspondant. Le fonctionnement de ces fonctions est simplifié pour ne pas trop alourdir le programme, et se scinde en deux temps : premier temps, vous cliquez sur le point ORIGINE de la fonction graphique, puis, deuxième temps, vous cliquez sur le point DESTINATION. Le programme dessine la fonction et revient au premier temps. Vous pouvez à tout moment revenir en mode PLOT en cliquant à droite.

Pour effacer le bloc, cliquez sur CLR, puis DEUX FOIS sur le dessin.
Le bouton G<-->D effectue une symétrie horizontale, le bouton H<-->B une symétrie verticale.
Vous pouvez mémoriser le bloc courant en cliquant sur STOCker. Le bloc s'affiche alors à côté de la zone zoomée. Pour rappeler le bloc, cliquez sur RECUpérer. Notez que la couleur zéro est transparente. Cette fonction est très pratique pour créer de nombreuse cases -presque- identiques!

Le bouton VOIR affiche le terrain avec les nouveaux blocs. Cliquez pour revenir à l'éditeur. Le bouton EDIT permet de retourner à l'édition du terrain.

EXPLICATION DU PROGRAMME

7000-7100 : Initialisation de l'écran et du menu.
7010 - définition des paramètres du ZOOM. Vous pouvez en modifier la taille en changeant le facteur de multiplication MZM et le monter jusqu'à 7. Attention, cela ralentit le programme!
7055 - dessin de la palette, et création des zones de test 50 à 65, une pour chaque couleur.
7060 - appel de la routine d'initialisation du mode PLOT.
7075 - copie et zoom du bloc courant.

7100-7200 : la boucle principale de l'éditeur, similaire à celle de l'éditeur de terrain (pourquoi changer ce qui fonctionne!).
7130 - la souris est ici aussi indiquée à l'aide d'un POLYMARKer, mais en en changeant régulièrement la couleur, pour éviter de la confondre avec la couleur de dessin.
7170 - on a cliqué! Une zone inférieure à 50 est une zone du menu.
7175 - la zone 100 correspond au ZOOM.
7180 - les zones comprises entre 50 et 99 sont les zones de la palette. Le calcul de la couleur pointée est une simple soustraction.
7185 - la zone 101, la dernière, permet de modifier le motif de remplissage. CPAT contient le motif courant, de 1 à 36.

7200-7250 : entrée des routines graphiques.
7200 - calcul de la coordonnée pointée par la souris.
7210 - un appui sur la touche de droite permet de sortir des fonctions graphiques.
7215 - ACT$ contient le libellé de la fonction graphique en action, ACTION contient le numéro de la ligne de traitement. Par ce Gosub transitent toutes les fonctions LINE, BOX, BAR, DISC etc.

7250-7300 : la routine de plot.
7250 - mise en route. ACTION pointera sur la ligne suivante. Cette ligne sera appelée chaque fois que vous cliquerez dans la zone de zoom.
7255 - la routine proprement dite. Nous devons dessiner chaque point 4 fois! La première fois, zoomée.
7260 - dessin du deuxième point (dans le bloc courant de gauche) et du troisième (dans le bloc courant de droite).
7265 - et le plus important, dans l'écran contenant les blocs.

7300-7305 : le zoom du bloc courant.

7310-7315 : sélection du bloc précédent / suivant.

Comme nous sommes malins (si, si!) nous appelons la routine de l'éditeur de terrain qui fait le plus gros travail pour nous. Il ne reste qu'à zoomer le nouveau bloc.

7325-7350 : dessin de la case d'information de du programme.

7325 - CPAT contient le motif de dessin courant, de 1 à 36. Or l'instruction SET PAINT désire DEUX paramètres: le premier indique un motif de points (2) ou de lignes (3), le second le numéro de ce motif. Cette ligne calcule les deux paramètres CPAT1 et CPAT2.

7340 - si un bloc est mémorisé, il est affiché à côté du zoom.

7350-7400 : routine générale de dessin. Par cette portion de code transitent les fonctions graphiques.

7355 - Initialisation de la première phase d'une fonction : on affiche la fonction (ACT$=ACT1$), et on fait pointer ACTION sur la ligne suivante. Rien de plus.

7360 - au prochain clic, le programme passera par cette ligne. Elle se borne à stocker le coordonnées de la souris dans des variables temporaires X1 et Y1, et à attendre le relâchement de la souris.

7365 - passage à la deuxième partie du la fonction en utilisant la même méthode qu'en 7355.

7370 - nous sommes en train de cliquer pour la deuxième fois. ACT contient l'adresse de la routine à appeler pour faire le dessin (ne pas confondre avec ACTION!). Par exemple pour une simple ligne, ACT est égal à 7400.

7380 - on recopie le bloc modifié dans l'écran de puzzle.

7385 - on zoome!

7400-7575 : les routines graphiques elles-mêmes.

7600-7605 : on revient à l'éditeur de terrain. Notez le POP qui dépile l'appel de la fonction elle-même. Oublier de le mettre constituerait le type même de "bug" difficile à déceler. S'il n'était pas présent, une adresse de RETURN serait empilée à chaque retour à l'éditeur. Il y a de fortes chances pour que rien ne se passe en utilisation normale du programme, le STOS acceptant jusqu'à 20 adresses empilées. Une utilisation intensive révèlerait ce bug : le magnifique terrain sur lequel vous travaillez depuis quatre heures serait perdu !
Puis, on se rebranche en 4025, pour une réinitialisation de l'écran.

Voilà pour aujourd'hui ! Maintenant il ne reste "qu'à" dessiner quelques dizaines de blocs! Dans le prochain article, nous reviendrons au jeu avec l'animation des ennemis... En attendant, STOSSEZ BIEN!

François LIONNET

PROGRAMMER EN OMIKRON (V)

Continuons notre "saga" avec l'exploration, ce mois-ci, des "Fonctions", puis nous passerons au graphisme et aux instructions qui l'accompagnent.

1.13. LES FONCTIONS

1.13.1. LES FONCTIONS MONOLIGNES

La fonction est en fait un cas particulier de la procédure. Son emploi est plus facile car la fonction est une procédure qui retourne toujours un seul paramètre. Voyons tout de suite ce que donne la procédure Mul (vue au chapitre précédent) en fonction :

```
DEF FN Mul(A,B,C,D)=A*B*C*D
```

Nous venons de définir la fonction qui calcule le produit de 4 nombres, une fois définie vous pouvez l'utiliser dans votre programme comme bon vous semble. Vous pourrez faire par exemple :

```
INPUT "Entrez quatre nombres: ";N,O,P,Q
PRINT FN Mul(N,O,P,Q)
A=FN Mul(N,O,P,Q)*FN Mul(N+1,O,O,Q)
PRINT A
END
DEF FN Mul(A,B,C,D)=A*B*C*D
```

Une fois la fonction Mul définie, l'expression **FN Mul(<Var>, <Var>, <Var>, <Var>)** se comportera exactement comme une variable, c'est-à-dire que vous pourrez l'utiliser partout où vous pouvez utiliser une variable qui ne serait que lue (car, en effet, on ne peut pas faire FN Mul(1,1,1,1)=1 ni d'ailleurs FN Mul(2,2,2,2)=458 !).

Il est bien sûr possible de faire une fonction avec des chaînes (à la place des variables numériques), comme par exemple :

```
DEF FN A$(nombre)="a"*Nombre
```

ici FN A$(5) sera égal à 'aaaaaa'

Exercice : réalisez une fonction qui aura une chaîne comme paramètre en entrée et qui nous retournera cette chaîne précédée d'un certain nombre d'espaces afin qu'elle soit centrée à l'écran (il s'agit ici en fait de faire la procédure Center vue précédemment en fonction).

Réponse :

```
DEF FN Center(Ac$)=SPACE$(80-LEN(Ac$)/2+Ac$
```

1.13.2. LES FONCTIONS MULTILIGNES

Il arrive fréquemment que le calcul du paramètre de retour soit beaucoup plus compliqué, donc que la fonction elle-même soit plus compliquée (qu'elle comporte des tests, des boucles, etc.). Dans ce cas, une seule ligne ne suffit pas et nous avons donc besoin des fonctions multilignes. Regardons la fonction Mul, vue précédemment, définie sur plusieurs lignes.

```
DEF FN Mul(A,B,C,D)
   Resultat=A*B*C*D
RETURN Resultat
```

ou encore :

```
DEF FN Mul(A,B,C,D)
RETURN A*B*C*D
```

Il existe deux différences avec les fonctions en une seule ligne :
--> il n'y a pas de signe égal après le DEF FN ;
--> on indique ce qu'il faut retourner après le RETURN ;

Une fonction monoligne du type :

```
DEF FN Nom(parametres)=expression
```

peut être écrite en multiligne comme ceci :

```
DEF FN Nom(paramètres)
RETURN expression
```

Exercice : faites une fonction qu'on appellera Code$, qui vous donnera 'code correct' si le paramètre est égal à 11170 et qui vous renverra 'code incorrect' sinon.

Réponse :

```
DEF FN Code$(code)
   IF Code=11170 THEN
      R$="code correct"
   ELSE
      R$="code incorrect"
   ENDIF
RETURN R$
```

Cette fonction s'appellera donc Code$, on l'appellera avec un paramètre numérique. Si ce paramètre est 11170, elle mettra dans R$ la chaîne 'code correct' sinon elle mettra dans R$ la chaîne 'code incorrect'. Enfin, elle retournera R$. C'est-à-dire que dans le programme **FN Code$(Nombre)** elle sera soit équivalente à **Code correct** soit équivalente à **Code incorrect** selon la valeur de Nombre.

1.14. ETIQUETTE, GOTO ET DATAS

Les étiquettes (ou labels) sont des repères que l'ont peut placer à n'importe quel endroit d'un programme et qui servent à le situer facilement. L'instruction GOTO permet que l'exécution du programme se poursuive à un autre endroit. Un exemple :

```
-Saisie_Nombre
INPUT "Entrez un nombre de 1 à 9: ";Nombre
IF Nombre<1 Or Nombre>9 THEN GOTO Saisie_Nombre
```

L'étiquette Saisie_Nombre indique l'endroit où se trouve la saisie.
Important : le nom d'une étiquette est toujours précédé d'un '-'. En effet, sans ce '-', il n'y a pas de différence entre une étiquette et un appel de procédure.
Important '2' : il faut prendre garde à ne pas mettre de '-' quand on se rend à l'étiquette. Par exemple, GOTO -Alpha n'est pas un saut correct, vous obtiendrez une erreur. Il faut faire GOTO Alpha.

L'instruction GOTO va entraîner un saut à l'instruction qui suit l'étiquette indiquée. Ici, GOTO Saisie_Nombre va entraîner l'exécution du INPUT à nouveau. Mais il fortement déconseillé d'utiliser trop de GOTO. Bien souvent, il existe un moyen plus simple que le GOTO. Dans notre exemple, un REPEAT...UNTIL serait en fait plus pratique.

En règle générale, nous vous conseillons d'utiliser au maximum les instructions structurantes, comme REPEAT...UNTIL, WHILE...WEND et les procédures, qui rendent le programme bien plus lisible. Rapidement, avec des GOTOS, vous ne saurez plus quelle partie de votre programme fait quoi, appelle quoi, et vous serez donc vite perdu !

L'instruction DATA, quant à elle, sert à inclure des données dans vos programmes. Voici un exemple :

```
RESTORE Nombres_Premiers
FOR I=1 TO 5
   READ A
   PRINT "Le nombre premier numéro ";I;" est ";A
NEXT I
-Nombres_Premiers
DATA 1,2,3,5,7
```

Étudions déjà cet exemple sans tenir compte du RESTORE et de l'étiquette Nombres_Premiers. L'instruction READ A va lire une donnée puis la stocker dans A. Cette donnée provient de la ligne DATA et la première donnée lue sera 1, la deuxième sera 2, etc., jusqu'à la cinquième qui sera 7.

On peut aussi lire des chaînes de caractères en les mettant entre guillemets dans la ligne de DATA. Il est aussi possible de mettre plusieurs lignes de DATA, par exemple la ligne DATA de notre exemple précédent est équivalente à :

```
DATA 1
DATA 2
DATA 3
DATA 5
DATA 7
```

Parlons maintenant du RESTORE et de l'étiquette. RESTORE sert à indiquer à l'ordinateur où se trouvent les DATAs à lire pour le prochain READ. En effet, le basic tient à jour un pointeur de DATA. Ce pointeur, au début, est situé sur la première donnée de la première ligne de DATA. Au fur et à mesure de la lecture, le pointeur "avance" dans les DATAS. Dans le cas où on a besoin de relire une donnée, il faut remettre le pointeur de DATA à la bonne place. Ce qui est réalisé par RESTORE.

Cause d'erreur : lorsqu'on exécute un READ A alors que la donnée en DATA est une chaîne, le basic vous indique qu'une erreur vient de se produire (confusion

de type). Idem si vous faites READ A$ alors que la donnée est numérique.

Si le pointeur de DATA est arrivé sur la dernière donnée et que vous tentez encore de lire une donnée, le basic affichera là aussi une erreur (Plus de DATA).

Conseils : quand des constantes se situent en DATA et que le programme va les consulter plus de deux fois, il est largement préférable de les charger dans un tableau une bonne fois pour toutes.

Dans les Basic moins évolués, il n'y a pas la possibilité de faire de labels, donc ils sont remplacés par des numéros de ligne, chaque ligne de programme a un numéro et GOTO et RESTORE sont suivis du numéro de la ligne concernée, ce qui est bien sûr beaucoup moins pratique. En Omikron vous pouvez, si vous le désirez, programmer avec des numéros de ligne (pour par exemple adapter des programmes basic venant d'autres machines), il suffit pour cela de ne pas taper CONTROL-ClrHome au début, et vous devrez alors numéroter vos lignes.
Voici un petit exemple de programme avec des numéros de ligne :

```
10 FOR I=1 TO 10
20   PRINT I
30 NEXT I
40 GOTO 10
```

1.15. LES GOSUB

GOSUB est une instruction qui était beaucoup (énormément !) utilisée dans les basics moins évolués (ne disposant pas de procédure ni de structure). En fait, le GOSUB n'est rien d'autre qu'une procédure sans passage de paramètre ni variable locale. Ce qui explique son moindre intérêt. GOSUB s'utilise de la manière suivante :

```
GOSUB Etiquette
```

Et le sous-programme se définit comme :

```
-Etiquette
    <instruction>
    <instruction>
    <instruction>
RETURN
```

Mais on préfèrera bien sûr une procédure :

```
DEF PROC Nom
    <instruction>
    <instruction>
      . . .
    <instruction>
RETURN
```

et l'appeler par :

```
Nom
```

C'est pourquoi un bon programmeur en Omikron ne servira jamais de GOSUB !

2. LE GRAPHISME

2.1. LE GRAPHISME SUR ORDINATEUR

Dessiner sur ordinateur est différent, par bien des aspects, de dessiner sur papier, car sur papier vous disposez d'un éventail infini de couleurs (en mélangeant la peinture), et la finesse de vos traits n'est limitée que par la finesse de votre crayon. Il n'en est pas de même sur ordinateur ! Sur le ST comme sur tous les autres ordinateurs, l'écran est divisé en un nombre limité de points (on appelle ces points des PIXELS) et chaque point peut prendre une couleur et une seule parmi 16 (en basse résolution). Ainsi, la largeur minimale d'un trait sera de 1 point, ce qui engendre bien entendu une multitude de contraintes.

De plus, nous avons vu que l'écran est divisé en un nombre limité de points, on peut donc associer à chaque point une coordonnée X (abscisse) et une coordonnée Y (ordonnée) permettant de le repérer. L'origine du repère est en haut à gauche, c'est-à-dire que le point de coordonnée X=0, Y=0 se trouve en haut à gauche de l'écran.

Le nombre de points n'est pas le même selon la résolution (moyenne et basse : moniteur couleur, haute : moniteur monochrome).
Basse résolution : moniteur couleur uniquement. Vous disposez de 320 points en largeur, de 200 points en hauteur et de 16 couleurs.
Moyenne résolution : moniteur couleur uniquement. Vous disposez ici de 640 points en largeur, de 200 points en hauteur et de 4 couleurs.
Haute résolution : moniteur monochrome uniquement. Vous disposez de 640 points en largeur, 400 points en hauteur mais seulement de deux couleurs, le blanc et le noir.

En basse résolution, nous avons vu que chaque point peut être d'une couleur parmi 16 couleurs. Cet ensemble de 16 couleurs s'appelle la "palette" et vous pouvez choisir votre palette de 16 couleurs parmi les 512 couleurs disponibles (4096 pour les STE). Vous ne pourrez pas avoir en même temps sur l'écran plus de 16 couleurs, soit les 16 couleurs de la palette. En moyenne résolution, le problème est exactement le même, mis à part le fait que la palette est réduite à 4 couleurs.

Une autre chose importante : il faut savoir que chacune des 512 couleurs du ST est composée d'une base de rouge, d'une base de vert et d'une base de bleu, chaque base allant de 0 à 7 (pour le STE, 0 à 15). "Illustrons" un peu : la couleur $700 (avec le rouge à 7, c'est-à-dire au maximum et le bleu et le vert à 0) sera du rouge vif, la couleur $703 (avec le rouge à 7, le vert à 0, le bleu à 3) sera une rouge violacé (en plus du rouge vient s'ajouter du bleu), etc.

On ne définit pas une couleur par un numéro arbitraire mais bien par un code de composition. A chaque combinaison correspond une couleur unique et à chaque couleur correspond une combinaison unique elle aussi.

2.2. DES INSTRUCTIONS !

Nous allons voir dans ce chapitre relativement long une multitude d'instructions graphiques qui vont vous servir à réaliser des effets graphiques.

2.2.1. DRAW

Cette instruction permet de tracer soit un point, soit une ligne d'un point de l'écran à un autre, voici comment l'utiliser :

DRAW <x>,<y> permet de tracer un point sur l'écran qui aura pour coordonnées x et y. Essayez de taper :

```
CLS:DRAW 100,100
```

Vous devez voir un petit point s'afficher sur l'écran, plutôt vers le haut et vers la gauche (le CLS sert à effacer l'écran avant de tracer le point).

DRAW <x>,<y> to <x2>,<y2> trace une droite du point de coordonnées x,y jusqu'au point de coordonnées x2,y2. Essayez de faire :

```
DRAW 100,100 TO 200,200
```

Remarque : il est aussi possible de mettre plusieurs TO afin de tracer plusieurs droites à la suite, essayez ce qui suit :

```
DRAW 100,100 TO 200,200 TO 300,100
```

DRAW TO <x>,<y> permet de tracer une droite du dernier point tracé jusqu'au point de coordonnées x,y.

Exemple : DRAW 100,100: DRAW TO 200,200

Exercice : Dessinez la figure ci-contre, sachant que ce carré fait 100 pixels de côté. Son coin supérieur gauche est le point de coordonnées 100,100. Ne regardez pas tout de suite la réponse qui suit, le problème est très simple...

Réponse :

```
DRAW 100,100 TO 200,100
DRAW TO 200,200
DRAW TO 100,200
DRAW TO 100,100
```

Evidemment, on aurait très bien pu faire cela en une seule instruction, avec un seul DRAW et plusieurs TO.

2.2.2. PALETTE

Comme son nom l'indique, cette instruction permet de changer la palette de couleurs. Voici comment elle fonctionne.

```
PALETTE <Couleur>,<Couleur>, . . .,<Couleur>
```

Le nombre de couleurs indiqué ne devra pas être supérieur au nombre de couleurs possible dans la résolution actuelle (2 en haute résolution, 4 en moyenne et 16 en basse). De plus, en haute résolution, il ne peut y avoir que le blanc ($777) ou le noir ($000). Ce qui limite donc la haute résolution a deux palettes possibles : $000,$777 ou $777,$000.

Le premier chiffre sera la couleur 1, le deuxième la couleur 2, etc. Mais vous n'êtes pas obligé d'indiquer toutes les couleurs de votre palette. Prenons le cas où vous ne voulez changer que les couleurs 2 et 3 de la palette en basse résolution. Il suffira de taper :

```
PALETTE ,Couleur2,Couleur3
```

La virgule indique que la valeur de la première couleur reste inchangée. Enfin, vos combinaisons déterminant les couleurs devront toujours être précédées du signe $ (nous allons vous expliquer pourquoi par la suite).
Exemple : (en basse ou en moyenne résolution)

```
PALETTE $700,$000,$050,$770
```

Cela aura pour effet de changer les 4 premières couleurs de la palette, et vous remarquerez que la première couleur correspond à celle du fond de l'écran (l'écran devient rouge).
NOTE : $ indique une valeur HEXADECIMALE (base 16).

Il nous reste encore beaucoup de choses à voir en Graphisme, mais ce sera pour la rentrée ! Allez, bonnes vacances...

Sébastien Enselme

LES GFA-PUNCHS DE L'ÉTÉ !

Basse résolution | GfA 2.0

Objet: D'incroyables scrolls différentiels!

Auteur: Olivier TARASSE | Note: A

```
1  C$=Mki$(&H333)+Mki$(&H700)+Mki$(&H500)+Mki$(&H700)+Mki$(
   &H300)+Mki$(&H700)+Mki$(&H500)+Mki$(&H700)+Mki$(&H200)+
   Mki$(&H700)+Mki$(&H500)+Mki$(&H700)+Mki$(&H300)+Mki$(&H700)+
   Mki$(&H500)+Mki$(&H700)
2  Void Xbios(6,L:Varptr(C$))
   T$="SALUT LES PUNCHISTES! SALUT TONTON! VOILA UN PUNCH QUI
   VA PLAIRE, J'ESPERE. DIS TONTON, CE QUI SERAIT SUPER, CA
   SERAIT DE PUBLIER UN PETIT ARTICLE SUR LA FACON D'ADAPTER UN
   PUNCH EN GFA 3.XX POUR EN FAIRE UN PUNCH EN GFA 2.XX. PARCE
   QUE NOUS, "
3  T$=T$+"LES PROGRAMMEURS EN 2.XX, ON A SOUVENT LES BOULES DE
   NE PAS POUVOIR FAIRE MARCHER DES PUNCHS QUI ONT L'AIR SUPER.
   JE ME DEMANDE VRAIMENT COMBIEN DE PUNCHISTES VONT TAPER CE
   TEXTE... AH, MONDE INGRAT...                              "
4  M$="BBJo"+Mki$(10)+"gd</"+Mkl$(535663)+Mkl$(320982)+
   Mkl$(&HE5EEFFF8)+Mkl$(&HE5EEFFF0)+Mkl$(&HE5EEFFE8)+Mkl$
   (&HE5EEFFE0)+Mkl$(&HE5EEFFD8)+Mkl$(&HE5EEFFD0)+Mkl$
   (&HE5EEFFC8)+Mkl$(&HE5EEFFC0)+Mkl$(&HE5EEFFB8)+Mkl$
   (&HE5EEFFB0)+Mkl$(&HE5EEFFA8)+Mki$(&HE5EE)
5  M$=M$+Mkl$(&HFFA0E5EE)+Mkl$(&HFF98E5EE)+Mkl$(&HFF90E5EE)+
   Mkl$(&HFF88E5EE)+Mkl$(&HFF80E5EE)+Mkl$(&HFF78E5EE)+Mkl$
   (&HFF70E5EE)+Mkl$(&HFF68E352)+Chr$(133)+"VBBM"+Mkl$
   (&HEEFF6051)+Mkl$(&HCEFFA64E)+Chr$(&H75)
6  M%=Varptr(M$)
7  For I%=0 To 4*320-1
8  Color Val("&H"+Mid$("9CD1",I% Div 320+1,1))
9  Line I% Mod 320,199,I% Mod 320,100+(I% Div 320+1)*15+Random
   ((I% Div 320+1)*3)+10*Cos((I% Mod 320)*Pi/20/(I% Div
   320+1))+10*Sin((I% Mod 320)*Pi/35*(5-(I% Div 320+1)))
10 Next I%
11 For S%=0 To 1000000
12 Void C:M%(L:Xbios(2)+31992,W:72,W:-1)
13 Void C:M%(L:Xbios(2)+31994,W:84,W:S% Mod 2=0)
14 Void C:M%(L:Xbios(2)+31996,W:96,W:S% Mod 4=0)
15 If S% Mod 16=0
16 Print At(40,8);Mid$(T$,(S% Div 16) Mod Len(T$)+1,1);
17 Endif
18 Void C:M%(L:Xbios(2)+10232,W:8,W:S% Mod 1=0)
19 Next S%
```

Basse résolution | GfA 3.0

Objet: Fait onduler votre écran.

Auteur: Michel GOUX | Note: B

```
1  SETCOLOR 15,&H777
2  SETCOLOR 0,0
3  DEFTEXT 1,0,0,6,0
4  TEXT 3,10,"-PLEASE PRESS A KEY TO SEE THE EFFECT-"
5  GET 0,4,319,10,texte$
6  CLS
7  FOR n%=0 TO 192 STEP 8
8  PUT 0,n%,texte$
9  NEXT n%
10 VOID INP(2)
11 OUT 4,18
12 REPEAT
13 PUT 30*SINQ(angle&),192,texte$
14 BMOVE ADD(XBIOS(2),1280),XBIOS(2),30560
15 VSYNC
16 ADD angle&,ADD(8,MUL(359,(angle&>350)))
17 UNTIL INP?(2)
18 OUT 4,8
19 SETCOLOR 15,0
20 SETCOLOR 0,&H777
```

Basse résolution | GfA 2.0

Objet: Un très beau serpent.

Auteur: Vincent GUESDON | Note: C

```
1  a$=STRING$(2000,4)
2  a%=V:a$
3  z$=STRING$(40,0)
4  zz%=V:z$
5  FOR f=0 TO 7
6  DPOKE zz%+(f*2),f*256+f*16+(7-f)
7  DPOKE zz%+14+(f*2),(7-f)*256+(7-f)*16+(f)
8  NEXT f
9  DPOKE zz%,0
10 REPEAT
11 BMOVE zz%+2,zz%+2-(2*(INT(r/2)=r/2)),30
12 DPOKE zz%+2,DPEEK(zz%+30)
13 DPOKE a%,x%
14 DPOKE a%+2,y%
15 VOID XBIOS(6,L:zz%)
16 MOUSE x%,y%,z%
17 BMOVE a%,a%+4,2000
18 INC r
19 PSET x%,y%,r MOD 15+1
20 PSET DPEEK(a%+1976),DPEEK(a%+1978),0
21 UNTIL LEN(INKEY$)
```

Toutes résolutions | GfA 2.0

Objet: Evite le RESET lors du changement de moniteur.

Auteur: J. ORLOFF | Note: B

```
1  PRINT "Une broche moniteur qui brinqueballe, ça vous fait
   des resets tout le temps..."
2  PRINT "Et c'est assez désagréable."
3  sys_base%=&H4F2 ! L'adresse en RAM-système contenant le
   début du système dans toutes les versions.
4  swv_vec%=&H46E  ! Contient l'adresse de la routine de
   changement de moniteur (RESET par défaut).
5  adresse%=LPEEK(sys_base%)
6  REPEAT
7    adresse%=adresse%+2
8  UNTIL DPEEK(adresse%)=&H4E75 ! On cherche le premier RTS en ROM.
9  SLPOKE swv_vec%,adresse%      ! On passe son adresse à swv_vec.
10 PRINT "Et voilà, vous êtes tranquilles jusqu'au prochain RESET!"
```

Basse résolution | GfA 2.0

Objet: Cassez tout: un Punch aggressif!

Auteur: Olivier TARASSE | Note: B

```
1  Setcolor 0,4,4,4
   V$=Mkl$(0)+Mkl$(7)+Mki$(4)+Mkl$(0)+Mkl$(0)+Mkl$(&H10000000)+
   Mkl$(&H18000000)+Mkl$(&H1C000000)+Mkl$(&H1FC0)+Mkl$(&H3FF0)+
   Mkl$(&H4F803046)+Mkl$(&HCF0030EF)+Mkl$(&H4E0031FC)+
   Mkl$(&H1C0003F0)+Mki$(&H3800)+String$(9,Mki$(0))
2  E$=Mkl$(0)+Mkl$(8)+Mki$(1)+String$(6,Mki$(0))+Mkl$(&H7E0)+
   Mkl$(&HFF0)+Mkl$(&H7E01818)+String$(3,Mkl$(&H660399C))+
   Mkl$(&HFF0300C)+Mkl$(&H3FFC)+Mkl$(&H3FFC)+Mkl$(&HFF00000)+
   Mkl$(&H7E00000)+Mkl$(&H3C01818)+Mkl$(&H180300C)
3  Do
4  K=Mousek
5  F=F+3-Vit/15+K*30
6  Print At(1,4);String$(20,"  "+Chr$(13)+Chr$(10));At(1,1);
   "VITESSE :";Left$(Str$((Vit+5)*1000),5);"   ";At(1,2);
   "ENERGIE :";Left$(Str$(800-F),5);" ";At(1,3);"DISTANCE:";
   300-D%;" ";String$(1-(Point(40,Y%+59)>1 Or Point(40,Y%+50)>
   1)*5000,0)
7  Prbox 36-300*(K-2),Y%+50,319-K+2,Y%+59
8  Inc D%
9  Sprite V$
10 Bmove Xbios(2)+160*25,Xbios(2)+160*25-8,170*149
11 Exit If Point(24,Y%+59)>1 Or Point(24,Y%+50)>1 Or F>800 Or D%>300
12 Sprite E$,319+15*(Random(2+D% Div 50)>0),48+Random(6)*16
13 Draw 319,183 To 312,150+Random(27) To 304,183 To 312,
   Random(27)+150 To 319,183 To 320,183 To 320,25 To 319,25 To
   312,27+Random(20) To 304,25
14 For I%=30 Downto Vit
15 Sprite V$,16,Y%+48
16 Y%=Y%+(Peek(&HFFFC02) Mod 4 Div 2=1)*(Y%<80)*(1+Vit Div 14)-
   (Peek(&HFFFC02) Mod 2=1)*(Y%>0)*(1+Vit Div 14)
17 Vit=Vit+(Peek(&HFFFC02) Mod 16 Div 8=1)*(Vit<30)*0.2-(Peek
   (&HFFFC02) Mod 8 Div 4=1)*(Vit>0)*0.2
18 Next I%
19 Loop
```

Basse résolution | GfA 3.0

Objet: Cycling de couleurs.

Auteur: Vincent GUESDON | Note: C

```
1  r$=STRING$(8192,0)
2  FOR u=0 TO 8192 STEP 64
3  DEFFILL INT(u/546)+1
4  PBOX INT(u/8192*320),0,INT(u/8192*320)+20,199
5  FOR i&=0 TO 7
6  DPOKE V:r$+u+i&*4,7-i&+&H8+u*240
7  DPOKE V:r$+u+i&*4+2,7-i&+u*240
8  DPOKE V:r$%+u+i&*4+32,i&+u*240
9  DPOKE V:r$+u+i&*4+34,i&+&H8+u*240
10 NEXT i&
11 NEXT u
12 s%=GEMDOS(32,L:0)
13 REPEAT
14 ad%=MOD(ad%+2,8000)
15 FOR t&=ad% TO ad%+64 STEP 2
16 BMOVE V:r$+t&,&HFF8242,30
17 NEXT t&
18 VSYNC
19 UNTIL LEN(INKEY$)
20 VOID GEMDOS(32,L:s%)
```

Basse résolution | GfA 2

Objet: Barre avec un scroll vitesse réglable

Auteur: Olivier TARASSE | Note: A

```
1  Setcolor 0,4,4,4
2  Setcolor 15,0,0,7
3  P$=String$(32000,0)
4  X%=Varptr(P$)
5  Void Xbios(5,L:X%,L:-1,W:-1)
6  Deftext 1,16,0,26
7  Draw 0,51 To 320,51 To 320,19 To 0,19
8  Text 0,112,"  - GFA PUNCH -"
9  Text 0,159,320,"CHOISIR LA VITESSE"
10 Text 0,190,"  AVEC LA SOURIS"
11 Repeat
12 A=A Mod 6+Mousex/2500
13 Bmove X%+21*160,X%+20*160,160*30
14 Bmove X%+80*160+I%*160,X%+50*160,160
15 Bmove X%+196*160,X%+52*160,160*4
16 Draw Mousex-7,52 To Mousex,55 To Mousex+7,52
17 Bmove X%+5*160,Xbios(2)+Int(((A-3)^2)*13)*160,80*160
18 Sound 1,-(Mousex>7)*(Int(((A-3)^2))+7)*-((A-3)<=0)*-(((A-3)
   ^2)>3),#125000/(Int(((A-3)^2)*13)+35)/1.5
19 I%=I% Mod 119+1
20 Until Inkey$<>""
```

Basse résolution | GfA 3.0

Objet: Apparition d'une image Degas

Auteur: Michel GOUX | Note: B

```
1  buffer$=STRING$(32034,CHR$(0))
2  FILESELECT "A:\*.PI1","",nm$
3  BLOAD nm$,VARPTR(buffer$)
4  VOID XBIOS(6,L:VARPTR(buffer$)+2)
5  OUT 4,18
6  FOR n|=0 TO 159
7  PBOX SUB(158,n|),99,ADD(159,n|),100
8  SOUND 1,12,3,1,0.5
9  NEXT n|
10 SOUND 1,13,2,1,2
11 FOR y|=99 DOWNTO 1
12 BMOVE ADD(VARPTR(buffer$),ADD(34,MUL(y|,160))),ADD(XBIOS(2),
   MUL(y|,160)),160
13 BMOVE ADD(VARPTR(buffer$),ADD(34,MUL(SUB(199,y|),160))),
   ADD(XBIOS(2),MUL(SUB(199,y|),160)),160
14 BOX 0,SUB(y|,1),319,SUB(200,y|)
   VSYNC
15 NEXT y|
16 SOUND 1,15,1,1,2
17 SOUND 1,0,0,0,0
18 VOID INP(2)
19 OUT 4,8
```

Basse résolution | GfA 3.0

Objet: 7 rouleaux à l'écran. Un Punch rotatif!

Auteur: Michel GOUX | Note: A

```
1  sens&=100
2  asm$=MKL$(&H42A73F3C)+MKL$(&H204E41)+MKL$(&H5C8F206F)+
   MKL$(&H4222F)+MKL$(&H8242F)+MKL$(&HC51CA)+MKL$(&HFFFE263C)+
   MKL$(&H63)+MKL$(&H33D800FF)+MKL$(&H824051CB)+MKL$(&HFFFE51C9)+
   MKL$(&HFFEE2F00)+MKL$(&H3F3C0020)+MKL$(&H4E415C8F)+
   MKL$(&H4E750000)
3  c1$="111222333444555666777666555444333222111011022033044055066
   077066055044033022011100200300400500600700600500400300200100"
4  c2$="010020030040050060070060050040030020010001002003004005 0
   060070060050040030020011102203304405506607706605504403302201
   101012023034045056067076065054043032021 01"
5  coul$=STRING$(182,CHR$(0))
6  asm%=V:asm$
7  FOR n|=0 TO 90
8  DPOKE VARPTR(coul$)+n|*2,VAL("&h"+MID$(c1$+c2$,n|*3+1,3))
9  NEXT n|
10 SETCOLOR 15,&H777
11 DEFTEXT 1,0,0,32
12 TEXT 65,100,"* GFA-PUNCH *"
13 OUT 4,18
14 REPEAT
15 ~C:asm%(L:VARPTR(coul$),L:90,L:v&)
16 VSYNC
17 ADD v&,sens&
18 sens&=MUL(sens&,((((v&<1) OR (v&>2000))+2)*2)-3)
19 UNTIL INP(-2)
20 OUT 4,8
```

Basse résolution | GfA 2.0

Objet: De superbes objets en 3D.

Auteur: Olivier TARASSE | Note: A

```
1  Dim X%(4),Y%(4)
2  For J%=1 To 5
3  Read C%,N%,F%,D%
4  C$=Mki$(&H444)+Mki$(7*C%)+Mki$(6*C%)+Mki$(3*C%)+Mki$(5*C%)
   +Mki$(2*C%)+Mki$(4*C%)+Mki$(C%)+String$(8,Mki$(0))
5  Void Xbios(6,L:Varptr(C$))
6  For Poly%=1 To N%
7  Read X1%,Y1%,L1%,H1%,X2%,Y2%,L2%,H2%
8  For A=2*Pi-D%/10 To -D%/10 Step -2*Pi/F%
9  For I%=0 To 4
10 X%(I%)=160+10*(-(X1%+L1%*Cos(A+2*Pi/F%*(I%=0 Or I%=4)))* (I%=0
   Or I%=1 Or I%=4)-(X2%+L2%*Cos(A+2*Pi/F%*(I%=3)))*(I%=2 Or I%=3))
11 Y%(I%)=50+10*(-(Y1%+H1%*Sin(A+2*Pi/F%*(I%=0 Or I%=4)))* (I%=0 Or
   I%=1 Or I%=4)-(Y2%+H2%*Sin(A+2*Pi/F%*(I%=3)))* (I%=2 Or I%=3))
12 Next I%
13 Deffill 7*(-Cos(A-D%/10)*(Cos(A-D%/10)>=0))+2
14 Polyfill 5,X%(),Y%()
15 Polyline 5,X%(),Y%()
16 Exit If Mousek
17 Next A
18 Next Poly%
19 Print String$(30000,0);Chr$(27);"E"
20 Next J%
21 Data 256,2,29,0,0,12,0,0,0,5,9,3,0,5,9,3,0,0,3,1,16,1,50,
   3,-8,-2,0,0,4,9,9,3,257,3,35,2,0,9,12,4,0,0,15,5,0,0,15,5,
   0,2, 6,2,0,2,6,2,0,-3,0,0,272,2,36,0,0,12,3,1,0,2,6,2,0,2,6,
   2,0,4,12,4, 273,3,31,0,0,9,9,3,0,9,3,1,0,9,3,1,0,5,3,1,0,4,
   9,3,0,-3,6,2
```

Basse résolution | GfA 3.0

Objet: Ondulation de texte.

Auteur: Michel GOUX | Note: C

```
1  DIM car$(13,27)
2  SLPOKE &HFF8240,&H777
3  FOR t|=0 TO 8
4  DEFTEXT 2,,,ADD(t|,13)
5  FOR n|=1 TO 27
6  TEXT 0,ADD(22,(t|/2)),MID$("SALUT A TOUS LES PUNCHISTES",n|,1)
7  GET 0,8,10,30,car$(t|,n|)
8  CLS
9  NEXT n|
10 NEXT t|
11 OUT 4,18
12 REPEAT
13 FOR n|=1 TO 27
14 PUT MUL(n|,11),100,car$(ABS(INT(SIN(car)*8)),n|)
15 ADD car,0.25
16 NEXT n|
17 UNTIL INP?(2)
18 SLPOKE &HFF8240,&H7770000
19 OUT 4,8
```

Basse résolution | GfA 3.0

Objet: Un "nouvel alphabet rigolo".

Auteur: J. ORLOFF | Note: C

```
1  facteur_esp=0.25
2  taille%=19
3  REPEAT
4  n%=INP(2)
5  FOR j%=0 TO 3
6  FOR k%=0 TO 1
7  @square(-((2^(k%+2*j%) AND n%)>0),x%+k%*taille%,y%+j%*taille%,
   -(-k% XOR EVEN(j%)))
8  NEXT k%
9  NEXT j%
10 x%=-(x%+(2+facteur_esp)*2*taille%<640)*(x%+INT((2+facteur_esp)
   *taille%))
11 y%=y%-(x%=0)*INT((4+facteur_esp)*taille%)
12 UNTIL n%=13
13 PROCEDURE square(bit%,x%,y%,i%)
14 i%=i% XOR bit%
15 DEFFILL i%
16 PBOX x%,y%,x%+taille%,y%+taille%
17 DEFFILL 1-i%
18 PCIRCLE x%+bit%*taille%,y%,taille%/2,(3-bit%)*900,(4-bit%)*900
19 PCIRCLE x%+(1-bit%)*taille%,y%+taille%,taille%/2,(1-bit%)
   *900,(2-bit%)*900
20 RETURN
```

Basse résolution | GfA 3.0

Objet: Multi Sprites!!!!!!!

Auteur: Olivier TARASSE | Note: B

```
1  Setcolor 0,4,4,4
2  Deffn X(D%,I)=160-(I-160)*(D=0 Or D=3)-(D=1 Or D=2 Or D=6)
   *(Cos(I/20)*150)-(D=3)*(Sin(I/4)*20)-(D=4)*(Cos(I/5)*I/5)-
   (D=5)*(Cos(I/30)*150)
3  Deffn Y(D,I)=100-(Sin(I/10)*65+Cos(I/30)*30)*(D=0 Or D=3 Or
   D=6)-(D=1 Or D=5)*(Sin(I/30)*80)-(D=2)*(Sin(I/25)*90)-(D=4)*
   (Sin(I/5)*I/7)-(D=5)*(Cos(I/4)*15)
4  Deffn L(D)=-(D=0)*150-(D=1)*120-(D=2)*120-(D=3)*16-(D=4)
   *99999-(D=5)*170-(D=6)*110
5  Defline 1,10,2
6  X%=Xbios(2)
7  Do
8  P$=String$(32500,0)
9  Bmove (Varptr(P$) Div 256)*256+256,X%,32000
10 Repeat
11 I=-(I+1)*(Inkey$="" And ((@L(D)<>99999) Or (I<816)))
12 Void Xbios(5,L:-((Varptr(P$) Div 256)*256+256)*((I Mod 2)=0)
   -X%*((I Mod 2)=1),L:-1,W:-1)
13 Color D+1
14 Plot @X(D,I) Mod 320,@Y(D,I)
15 Color 0
16 Plot @X(D,I-@L(D)) Mod 320,@Y(D,I-@L(D))
17 Void Xbios(5,L:-1,L:-((Varptr(P$) Div 256)*256+256)*((I Mod
   2)=0)-X%*((I Mod 2)=1),W:-1)
18 Until Gemdos(11)
19 D=(D+1) Mod 7
20 Loop
```

Basse résolution | GfA 2.0

Objet: Des sprites sur une image Degas PI1...

Auteur: Vincent GUESDON | Note: C

```
1  DIM spr$(9)
2  FOR m=1 TO 150 STEP 16
3  PCIRCLE m+8,8,ABS(-5+(m/8) MOD 10)+3
4  spr$(INT(m/16))=MKI$(0)+MKI$(0)+MKI$(1)+MKI$(0)+MKI$(4)
5  FOR g=0 TO 15
6  FOR f=-1 TO 15
7  mot=-(mot-(2^(15-f))*(PTST(f+m,g)<>0))*(f<>-1)
8  NEXT f
9  spr$(INT(m/16))=spr$(INT(m/16))+MKI$(mot)+MKI$(mot)
10 NEXT g
11 NEXT m
12 FILESELECT "A:*.*"," ",n$
13 BLOAD n$,XBIOS(2)-34
14 VOID XBIOS(6,L:XBIOS(2)-32)
15 REPEAT
16 t=t+INT(f%/9)
17 f%=MOD(f%+1,10)
18 SPRITE spr$(f%),(f%*66) MOD 300,100+SIN((t+f%*2)/18*PI)*80
19 UNTIL MOUSEK=2
```

Basse résolution | GfA 3.0

Objet: Générateur de fugues. Un Punch cacophonique!

Auteur: J. ORLOFF | Note: B

```
1  DIM mesp(32,2)
2  DEFFN h(deg%)=modul%+ASC(MID$("acdfhil",deg% MOD 7+1))-ASC("a")
   +12*(deg% DIV 7) ! Hauteur chromatique de la note, en
   fonction de son degré dans la gamme modulée par modul%
3  REPEAT
4  modul%=modul%+7*((RND>0.4 AND modul%>=0)-(RND>0.4 AND modul%
   <=0)) ! Modulation d'une quinte
5  FOR j%=1 TO 16 ! Changer la mesure ici.
6  temp%=degre%
7  change%=-(RND>(0.2+0.1*(j%=1)-0.2*(0<>INSTR("135",CHR$(ASC
   ("0")+temp% MOD 7)))))
8  REPEAT ! Attend une touche ou une "harmonie"
9  degre%=temp%+(RANDOM(2)+1)*(2*RANDOM(2)-1+3*(temp%>40)-3*(
   temp%<25))
10 UNTIL INKEY$<>"" OR 0=INSTR("bcglns",CHR$(ASC("a")+ABS(@h
   (degre%)-mesp(j%,1))))
11 degre%=temp%+change%*(degre%-temp%)
12 SOUND 4-2*mesp(j%,2),12-1*(j%=1),mesp(j%,1) MOD 12,mesp(j%,1)
   DIV 12
13 SOUND 4-3*change%,13-1*(j%=1),@h(degre%) MOD 12,@h(degre%) DIV 12
14 WAVE 3
15 PAUSE 6  ! Changer le tempo ici.
16 mesp(j%,1)=@h(degre%+0) ! Essayez +/- 2 ou +/- 4
17 mesp(j%,2)=change%
18 NEXT j%
19 UNTIL INKEY$<>""
```

LE STOS-PUNCH

Ce Stos-punch est l'oeuvre de Fabien Potencier, de Saint Venant, il gagne quelques softs de la Boutique Pressimage. Après avoir chargé la banque MAP.MBK et après avoir branché le joystick dans le port 2, il ne vous reste plus qu'à avaler le plus possible de pastilles ou de goodies! Un Stos Punch, c'est un listing Punch de 10 lignes maximum, que vous envoyez à Tonton Stos sur disquette (impossible de vous la renvoyer!) accompagnée d'un courrier dans lequel vous nous direz quels softs Pressimage vous feraient plaisir...

```
1 restore 9 : for I=1 to 114 : read F : next : hide : key off : mode 0 :
  curs off : windopen 1,0,0,40,12,0,3 : for I=1 to 2 : read A$ : wait 90
  : fade 3 : wait 21 : clw : locate 0,6 : centre A$ : wait 20 : fade
  3,,$777,$0 : wait 21 : next : restore 10 : for I=1 to 114 : read F :
  next : for I=1 to 2 : read A$ : wait 90 : fade 3 : wait 21 : clw :
  locate 0,6 : centre A$ : wait 20 : fade 3,,$777,$0 : wait 21 : next :
  wait 90 : fade 3 : wait 21 : clw : windel 1 : TB=2 : dim A(20,10)
2 restore 9-TB : mode 0 : key off : for I=1 to 16 : sprite 1,16+16*I,48,
  23 : wait vbl : put sprite 1 : next : for I=1 to 16 : sprite 1,16+16*I,
  176,23 : wait vbl : put sprite 1 : next : sprite 1,16,48,22 : wait vbl
  : put sprite 1 : sprite 1,288,48,24 : wait vbl : put sprite 1 : sprite
  1,16,176,19 : wait vbl : put sprite 1 : sprite 1,288,176,19 : wait vbl
  : put sprite 1 : for I=1 to 8 : sprite 1,16,48+16*I,20 : wait vbl : put
  sprite 1 : next : for I=1 to 8 : sprite 1,288,48+16*I,20 : wait vbl
3 put sprite 1 : next : J=0 : X=16 : Y=7 : X1=16 : Y1=7 : windopen 1,3,
  0,35,3,3,3 : pen 1 : while J<7 : J=J+1 : for I=1 to 16 : read A :
  A(I,J)=A : sprite 1,16+16*I,48+16*J,A(I,J) : wait vbl : put sprite 1 :
  next : wend : read TPS : read SS : fade 29,$0,$333,$555,$777,$330,$3,
  $5,$7,$550,$300,$500,$700,$770,$30,$50,$70 : curs off : locate 9,0 :
  input "NIVEAU (1->5) :";N : TPS=TPS/N : clw : while joy=0 : print :
  locate 10,0 : print "R E A D Y !"; : wend : print : locate 2,0 : print
  "SCORE : "; : locate 18,0 : print "TEMPS : ";
4 clear key : if jleft=-1 and X>1 and A(X-1,Y)<>10 then X=X-1 else if
  jright=-1 and X<16 and A(X+1,Y)<>10 then X=X+1 else if jup=-1 and Y>1
  and A(X,Y-1)<>10 then Y=Y-1 else if jdown=-1 and Y<7 and A(X,Y+1)<>10
  then Y=Y+1
5 sprite 2,16+16*X1,48+16*Y1,13 : wait vbl : put sprite 2 : off : sprite
  1,16+16*X,48+16*Y,7 : TPS=TPS-1 : locate 25,0 : print "    "; : locate
  25,0 : print TPS; : if A(X,Y)<>13 then SC=SC+A(X,Y)*10*N : A(X,Y)=13 :
  X1=X : Y1=Y : SS=SS-1 : shoot : locate 10,0 : print SC;
6 if TPS=0 then bell : bell : bell : clw : pen 14 : locate 10,0 : print
  "P E R D U !"; : clear key : wait key : run else if SS=0 then for I=1
  to TPS-2 step 3 : TPS=TPS-3 : SC=SC+3*N : bell : locate 10,0 : print
  SC; : locate 25,0 : print "    "; : locate 25,0 : print TPS; : next :
  if TB<>-1 then TB=TB-1 : goto 2 else clw : locate 0,0 : pen 11 : print
  "VERY GOOD!! "; : pen 14 : print " YOUR SCORE : ";SC; : clear key :
  wait key : run else wait vbl : clear key : goto 4
7 data 1,2,3,4,5,5,5,5,1,1,2,2,4,3,5,1,10,10,10,10,2,10,10,10,2,10,10,
  10,1,10,10,4,10,9,2,1,2,10,11,10,5,10,11,10,9,10,11,10,10,2,10,10,4,
  10,11,10,2,10,11,10,5,10,11,10,10,2,6,10,3,10,11,10,3,10,11,10,3,10,11,
  10,10,10,10,10,8,10,10,10,4,10,10,10,3,10,10,2,1,2,3,4,5,4,3,2,1,2,3,
  4,5,4,3,13,500,54
8 data 6,2,10,10,5,10,10,10,1,2,5,4,3,2,1,10,10,3,10,1,5,4,2,10,1,3,2,3,
  2,1,10,2,2,3,10,10,2,10,2,1,3,2,5,2,4,10,1,2,2,3,5,10,25,10,2,1,10,10,
  2,3,10,2,3,4,2,3,1,10,10,10,2,3,10,10,2,10,5,2,1,3,10,10,2,5,2,3,1,2,
  9,2,10,1,2,3,5,2,11,11,10,2,2,3,2,5,2,1,2,4,2,3,2,13,740,80
9 data 10,10,2,3,5,2,1,6,2,3,2,10,11,11,11,11,25,3,10,10,2,3,5,10,2,10,
  2,10,11,11,11,11,10,2,3,10,5,6,2,10,2,10,4,10,11,11,11,11,3,10,2,9,10,
  8,1,10,5,10,8,10,10,10,10,10,2,3,10,2,5,10,2,10,2,3,10,2,1,3,5,2,2,3,1,
  10,2,10,2,10,2,1,3,1,2,10,10,10,2,1,5,4,2,3,2,10,1,2,3,4,5,2,1,13,600,
  66,"POTENCIER Fabien","présente"
10 data 1,1,1,1,1,1,1,1,1,2,3,1,5,4,9,2,10,10,10,10,2,10,1,2,3,10,2,
  10,10,2,3,5,10,6,1,2,3,10,10,25,2,10,3,10,11,10,2,1,10,10,3,3,2,10,25,
  10,2,10,4,10,11,10,2,3,10,2,1,2,3,10,2,3,10,10,2,10,11,10,2,3,10,10,10,
  10,5,10,2,1,3,10,5,10,10,2,3,1,2,5,5,3,1,2,8,1,3,2,2,1,2,1,13,700,73,
  "BOUFFE-TOUT","Juin 1990"
```

EMULATEUR VIDEOTEX

Plus rapide qu'Emulcom, plus beau que ZZ-Com, en exclu-lulu, voici les listings, en GfA et en Assembleur, qu'Emmanuel Anne a écrits pour son émulateur ! C'est en basse résolution et c'est pas chez Ardisson !

Les fans de listings seront ravis, puisqu'il y a 13 pages, le record est battu. Les fans de Vidéotex seront rassasiés puisque tout y est, ou presque. Il manque juste le masquage et le clignotement (que vous pourrez cependant retrouver dans une version plus étoffée que celle-ci, commercialisée dans les prochains mois à la Boutique de Pressimage), tout le reste y est ! Et c'est rapide, et le résultat à l'écran est lisible.

L'émulation elle-même est programmée en Assembleur (en GfA-Assembleur, pour être plus précis), et le contrôle, l'environnement de l'émulation est assurée par les routines GfA. D'autre part, un autre listing GfA s'occupe de générer la fonte écran.

Vous avez 3 listings Assembleur, qui ont été prévus pour le GfA-Assembleur, et qui doivent tous les 3 être intégrés aux INLINE du source GfA, l'un d'eux après l'avoir modifié à l'aide du deuxième listing GfA, comme on vous l'explique juste avant ce dernier (mode d'emploi du jeu de piste en page 152).

Avec cette version, on peut soit émuler le Minitel, soit relire un fichier Vidéotex sauvegardé (vous consultez un serveur, et vous capturez les écrans affichés pour les visualiser ultérieurement). Pour revenir au menu principal, utilisez <F2>. Pour quitter l'émulateur, tapez <F9>. Les commandes de l'émulateur sont indiquées dans le source GfA.

Comme tout document publié dans ST Magazine, toute utilisation commerciale ou reproduction, même partielle, de ce listing est interdite sans l'accord de l'auteur et de Pressimage. ST Magazine se réserve le droit d'engager des poursuites en cas de non-respect de cette interdiction.

Le tout est sur la disquette ST Mag 43 !

LISTING GFA 1

```
' Emulateur minitel version 3.4 modifiée Juin 1990
' (Uniquement couleur)
RESERVE 150000
' Afin de pouvoir utiliser des buffers par MALLOC
DEFWRD "a-z"
resol=XBIOS(4)
IF resol<>0
  ALERT 3,"Cette version ne tourne qu'en|basse...",1,"Désolé",b!
  RESERVE
  END
ENDIF
'
' Initialisation de la palette...
'
RESTORE palette
FOR n=0 TO 7
  READ h
  p$=p$+MKI$(h)
NEXT n
p$=p$+p$
~XBIOS(6,L:V:p$)
palette:
DATA $000,$700,$070,$770,$007,$707,$077,$777
~XBIOS(15,7,0,174,-1,-1,-1)
DEFMOUSE 0                    ! Pour le compilateur
'
' *******************
' * Routines INLINE *
' *******************
'
INLINE adr%,7316
INLINE instr%,34
INLINE instr2%,38
'
' Inits (Certaines variables se trouvent
' directement dans la routine...)
'
DEFFN state(n)=BYTE{adr%+24*n+11}
DIM r%(15),page%(500)
'
y$=SPACE$(2)          ! Ca, ça servira à recevoir les coords du locate
taille_ecran%=40*1280 ! 40 lignes
taille_rs%=2048       ! Taille du tampon alloué pour la RS232
buff=100
' Mettre ici la place à reserver, en Ko, pour le buffer pages
sensible=80
' longueur minimum d'une page - important pour la lecture de fichier
init%=adr%+&H1C
$C+
ABSOLUTE phys%,C:init%()
$c
ABSOLUTE coordx|,V:phys%+4
ABSOLUTE coordy|,V:coordx|+1
ABSOLUTE color|,V:coordy|+1
ABSOLUTE attrib|,V:color|+1
ABSOLUTE fond|,V:attrib|+1
ABSOLUTE encre|,V:fond|+1
ABSOLUTE memd3|,V:encre|+1
ABSOLUTE mematt|,V:memd3|+1
ABSOLUTE physmin%,V:mematt|+1
ABSOLUTE physmax%,V:physmin%+4
ABSOLUTE physmed%,V:physmax%+4
phys_alloc%=MALLOC(32000+taille_ecran%+256)
physmin%=(phys_alloc%+&H100) AND &HFFFFFF00
IF physmin%<&H200
 ALERT 2,"Erreur dans les allocations|mémoire... désolé.",1,"ABANDON",b!
  gem_palette
  END
ENDIF
physmax%=physmin%+taille_ecran%
physbase%=XBIOS(2)
phys%=physmin%+SHR(taille_ecran%,1)
physmed%=phys%
~XBIOS(5,L:phys%,L:phys%,-1)
main%=adr%+&H1E
newx%=adr%+&H20
curseur%=adr%+&H24
vide
path$=CHR$(GEMDOS(25)+65)+":"+DIR$(0)
alloc_rs232
ressource   ! Dans cette version, ce n'est pas une ressource !
r%(4)=FALSE ! Mode rouleau
r%(0)=1     ! coordonnées curseur
r%(1)=2
r%(3)=FALSE ! Fonte texte, aucun attribut.
CLR no%     ! Nombre d'octets traités
fond|=0
encre|=7
'
r%(2)=12
RCALL main%,r%()
ON BREAK GOSUB fin
'
' *****************
' Boucle principale
' *****************
'
start:
CLR exit
REPEAT
  ' Select/Case sur des mots, optimiser le tps d'execution
  $S&,S
  SELECT @traite
  CASE &H1F
  ' locate
    y2=@traite
    IF y2<32
      erreur!=TRUE
    ELSE
      x2=@traite
      IF x2<32
        erreur!=TRUE
      ELSE
        BYTE{V:y$}=y2
        BYTE{V:y$+1}=x2
        IF VAL?(y$)=2
          y2=ADD(VAL(y$),1)
          x2=1
        ELSE
          SUB y2,63
          SUB x2,64
        ENDIF
        IF x2>0 AND x2<41 AND y2>0 AND y2<26
          IF r%(7)
            RCALL curseur%,r%()
          ENDIF
          IF r%(1)>1
            coordx|=r%(0)
            coordy|=r%(1)
            memd3|=r%(3)
            mematt|=attrib|
            color|=encre|
          ENDIF
          r%(0)=x2
          r%(1)=y2
          IF r%(7)
            RCALL curseur%,r%()
          ENDIF
          r%(3)=FALSE
          attrib|=&H80
          encre|=7
        ENDIF
```

```
      ENDIF
    ENDIF
  CASE &H12
  ' répète le dernier caractère
    r%(6)=@traite-65
    r%(2)=vieux
    IF r%(6)>=0
      RCALL main%,r%()
    ELSE
      IF r%(6)<-33
        erreur!=TRUE
      ENDIF
      r%(6)=0
    ENDIF
  CASE 27
    ' Codes ESCAPE
    IF @traite=&H3A
      ' Pro2 (le protocole pas le prg !)
      IF @traite=115
        ' Acquittement statut fonctionement...
        @mode(@traite)
      ELSE IF c<32
        erreur!=TRUE
      ENDIF
    ELSE IF c=&H23 OR c=&H20
      IF @traite<32
        erreur!=TRUE
      ELSE IF @traite<32
        erreur!=TRUE
      ELSE
        IF buffer!
          no%=SUB(no%,4)
        ENDIF
      ENDIF
    ELSE IF c=&H3B OR c=&H21
      IF @traite<32
        erreur!=TRUE
      ELSE IF @traite<32
        erreur!=TRUE
      ELSE IF @traite<32
        erreur!=TRUE
      ELSE
        IF buffer!
          no%=SUB(no%,5)
        ENDIF
      ENDIF
       ' Signification des bits de D3
      ' 0 : Mode texte/graphique
      ' 1 : masquage caractères graphiques
      ' 2 : Double largeur
      ' 3 : Double hauteur
      ' 4 : Inversion vidéo
    ELSE IF c>=&H40 AND c<=&H47
      encre|=c-&H40
    ELSE IF c>=&H50 AND c<=&H57
      attrib|=OR(AND(attrib|,&H70),c-&H50)
    ELSE IF c=91
      ' Déplacement curseur - Minitel 1B
      IF r%(7)
        RCALL curseur%,r%()
      ENDIF
      IF @traite>31
        SELECT c
        CASE 65
          IF r%(1)>2
            DEC r%(1)
          ENDIF
        CASE 66
          IF r%(1)<25
            INC r%(1)
          ENDIF
        CASE 67
          IF r%(0)<40
            INC r%(0)
          ENDIF
        CASE 68
          IF r%(0)>1
            DEC r%(0)
          ENDIF
        ENDSELECT
      ELSE
        erreur!=TRUE
      ENDIF
      IF r%(7)
        RCALL curseur%,r%()
      ENDIF
    ELSE IF c=90
      ' Soulignement ou masquage
      IF BTST(r%(3),0)
        ' à distance seulement en mode texte
        r%(3)=r%(3) OR 2
      ELSE
        attrib|=BSET(AND(attrib|,&H3F),6)
      ENDIF
    ELSE IF c=89
      ' suppression du soulignement/masquage
      IF BTST(r%(3),0)
        r%(3)=BCLR(r%(3),1)
      ELSE
        attrib|=AND(attrib|,&H3F)
      ENDIF
    ELSE IF c=78   ! Double largeur
      r%(3)=r%(3) OR 4
      r%(3)=r%(3) AND &HF7
    ELSE IF c=77   ! Double hauteur
      r%(3)=r%(3) OR 8
      r%(3)=r%(3) AND &HFB
    ELSE IF c=76   ! Taille normale
      r%(3)=r%(3) AND &HF3
    ELSE IF c=79   ! Double taille
      r%(3)=r%(3) OR 12
    ELSE IF c=92   ! Inversion vidéo OFF
      r%(3)=BCLR(r%(3),4)
    ELSE IF c=93   ! Inversion vidéo
      r%(3)=BSET(r%(3),4)
    ELSE IF c=72 OR c=73 ! clignotement ON/OFF
    ELSE IF c=94   ! code inconnu !!!
    ELSE IF (c>&H34 AND c<&H3A) OR c=&H25
OR c=&H2F OR c=&H22
      IF @traite<32
        erreur!=TRUE
      ELSE
        IF buffer!
          no%=SUB(no%,3)
        ENDIF
      ENDIF
    ELSE IF c<32
      erreur!=TRUE
    ELSE IF c<=95
      attrib|=BCLR(attrib|,7)
    ENDIF
  CASE &H16,&H19  ! Accents
    IF @traite>31
      SELECT c
      CASE 35
        r%(2)=132
      CASE 36
        r%(2)=36
      CASE 38
        r%(2)=0
      CASE 39
        r%(2)=157
      CASE 44
        r%(2)=128
      CASE 45
        r%(2)=131
      CASE 46
        r%(2)=129
      CASE 47
        r%(2)=130
      CASE 48
        r%(2)=133
      CASE 49
        r%(2)=150
      CASE 56
        r%(2)=151
      CASE 60
        r%(2)=152
      CASE 61
        r%(2)=153
      CASE 65,66,67,72,75
        t=c
        c=@traite
        accents
      CASE 106
        r%(2)=154
      CASE 122
        r%(2)=155
      CASE 123
        r%(2)=156
      DEFAULT
        r%(2)=95
      ENDSELECT
      RCALL main%,r%()
    ELSE
      erreur!=TRUE
    ENDIF
  CASE &H13
    IF @traite=&H56
      ' changement de mode fonctionement
      IF minitel!
        REPEAT
          ' on attend l'ack de chgt de mode
          inp$=inp$+INPAUX$
          'sinon Minitel et ST ne font pas le
          r=INSTR(inp$,"s")
          'changement de mode en même temps...
        UNTIL r>0
        '
        IF buffer!
          ' on le met dans le buffer
          BYTE{buffer%+no%-2}=27
          ' à la place du changement de mode
          BYTE{buffer%+no%-1}=58
          BYTE{buffer%+no%}=115
          INC no%
        ENDIF
        IF LEN(inp$)=r+2
          c=INP(1)
          ' on l'interprète
          inp$=LEFT$(inp$,PRED(r))
        ELSE
          ' ce cas doit être rare compte tenu
          ' de la vitesse du ST...
          c=ASC(MID$(inp$,r+3))
          inp$=LEFT$(inp$,PRED(r))+MID$(inp$,r+4)
        ENDIF
        IF buffer!
          BYTE{buffer%+no%}=c
          INC no%
        ENDIF
        @mode(c)
' En fait, c'est ici qu'on fait vraiment
' le changement de mode
        IF buffer!
' pour un gain de vitesse...
```

```
        BMOVE V:inp$,buffer%+no%,LEN(inp$)
        ADD no%,LEN(inp$)
      ENDIF
    ELSE
      ' Dans le cas où on lit un fichier
      mem%=no%
      ' on cherche tout de suite l'ack
      REPEAT
        ' Dans le cas d'une lecture d'un fichier
        INC mem%
        ' créé avec cette version, on ne sera jamais
        ' dans ce cas. Cette manip est nécessaire pour effectuer le
        ' changement de mode en même temps que le Minitel
      UNTIL (BYTE{ad%+mem%-3}=27 AND BYTE{ad%+mem%-2}=58 AND
BYTE{ad%+mem%-1}=115) OR SUB(mem%,no%)>256
      IF SUB(mem%,no%)<=256
        @mode(BYTE{ad%+mem%})
      ENDIF ! Si l'acquittement n'est pas trouvé, changement de mode impossible
    ENDIF
  ELSE IF c=&H5E OR c=&H53 ! Reset vidéotex
    r%(3)=0
    r%(4)=0
    IF r%(7)
      RCALL curseur%,r%()
    ENDIF
    r%(7)=0
    ~BIOS(11,16) ! MAJUSCULES
  ELSE IF c<32
    erreur!=TRUE
  ENDIF
CASE 7 ! Bell
  PRINT CHR$(7);
DEFAULT
  r%(2)=c
  IF c>31
    vieux=c
  ENDIF
  RCALL main%,r%()
ENDSELECT
fin:
UNTIL exit>0
SELECT exit
CASE 9
  ALERT 3," Vraiment quitter le |     programme ?",1,"OUI|NON",exit
  IF exit=1
    fin
  ENDIF
CASE 2
  ressource
ENDSELECT
GOTO start
'
' **********
' Procedures
' **********
'
PROCEDURE fonctionement
  ' Status fonctionnement ?
  OUT 1,&H1B,&H39,&H72
RETURN
PROCEDURE mode(c)
  ' Place le ST dans le mode renvoyé par fonctionement
  IF BTST(c,1)
    r%(4)=-1
  ELSE
    r%(4)=0
  ENDIF
  IF BTST(c,3)
    VOID BIOS(11,0)
  ELSE
    ~BIOS(11,16)
  ENDIF
RETURN
PROCEDURE vide
  WHILE INP?(1)
    ~INP(1)
  WEND
RETURN
PROCEDURE accents
  SELECT t
  CASE &H41
    SELECT CHR$(c)
    CASE "a"
      r%(2)=134
    CASE "e"
      r%(2)=135
    CASE "u"
      r%(2)=136
    DEFAULT
      r%(2)=c
    ENDSELECT
  CASE &H42
    IF c=69 OR c=101
      r%(2)=137
    ELSE
      r%(2)=c
    ENDIF
  CASE &H43
    IF c=65 OR c=97
      r%(2)=138
    ELSE IF c=69 OR c=101
      r%(2)=139
    ELSE IF c=111 OR c=79
      r%(2)=140
    ELSE IF c=73 OR c=105
      r%(2)=141
    ELSE IF c=85 OR c=117
      r%(2)=142
    ENDIF
  CASE &H48
    IF c=69 OR c=101
      r%(2)=143
    ELSE IF c=65 OR c=97
      r%(2)=144
    ELSE IF c=73 OR c=105
      r%(2)=145
    ELSE IF c=111 OR c=79
      r%(2)=146
    ELSE IF c=85 OR c=117
      r%(2)=147
    ENDIF
  CASE 75
    IF c=67 OR c=99
      r%(2)=148
    ENDIF
  DEFAULT
    r%(2)=c
  ENDSELECT
RETURN
FUNCTION traite
' envoie un code au prg en provenance du Minitel ou d'un fichier
  LOCAL car,t
  WHILE car=0 AND (NOT erreur!)
    IF minitel!
      IF LEN(inp$)
        car=BYTE{V:inp$+no2%}
        INC no2%
        IF no2%=LEN(inp$)
          CLR no2%,inp$
        ENDIF
      ELSE IF INP?(1)
        car=INP(1)
        IF buffer!
          BYTE{ADD(buffer%,no%)}=car
```

```
        INC no%
        IF no%>long_buffer%
          ALERT 1," Capacité du buffer | dépassée !!! sauvegarde |
immédiate... ",1,"Compris",b!
          CLR car
          KEYPRESS 19
        ENDIF
      ENDIF
    ENDIF
    IF INP?(2)
      @touche
      EXIT IF exit
    ENDIF
  ELSE
    '
    ' ********************
    ' Lecture d'un fichier
    ' ********************
    '
    car=BYTE{ad%+no%}
    IF no%<long%
      INC no%
    ELSE
      car=1 !CODE BIDON
    ENDIF
    '
    ' Pression d'une touche pendant l'affichage
    '
    IF INP?(2)
      t=INP(2)
      IF t=200
        ' Retour
        IF page>0
          DEC page
          update_page
        ENDIF
      ELSE IF t=208
        IF page<PRED(nb_pages)
          INC page
          update_page
        ENDIF
      ELSE IF t=159
        r%(4)=NOT r%(4)
      ELSE IF t>=187 AND t<=196
        exit=t-186
      ENDIF
    ENDIF
    '
    ' Gestion de l'attente d'une touche
    '
    IF efface>0
      DEC efface
    ELSE
      IF no%=page%(SUCC(page))
        IF sortie
          imprime
        ELSE
          PRINT AT(34,1);SUCC(page);"/";nb_pages;" ";
          DEC no%
          CLR car
          REPEAT
            t=INP(2)
          UNTIL t=208 OR t=200 OR t=159 OR (t>=187 AND t<=196)
          IF t=200        ! Retour
            IF page>0
              DEC page
              update_page
            ENDIF
          ELSE IF t=208
            IF page<PRED(nb_pages)
              INC page
              update_page
```

```
            ENDIF
          ELSE IF t=159
            r%(4)=NOT r%(4)
          ELSE IF t>=187 AND t<=196
            exit=t-186
            erreur!=TRUE
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
  '
  ' L'entrée des données a maintenant été traitée
  '
WEND
IF erreur!
  erreur!=FALSE
  RETURN c
ELSE
  c=car
  RETURN car
ENDIF
ENDFUNC
PROCEDURE touche
  ' Commandes :
  ' HELP      : Guide
  ' UNDO      : Sommaire
  ' BACKSPACE : Correction
  ' Delete    : Anulation
  ' Ctrl C    : Connexion/Fin
  ' Ctrl S    : Sauvegarde
  ' Ctrl F    : Déconexion
  ' Ctrl R    : Répétition
  ' Alt+num   : Code ASCII num ("&" hexa ou "" décimal)
  ' Alt+S     : Scrolling ON/OFF
  ' Tous les accents sont aussi disponibles, comme sur tout
  ' bon traitement de textes...
  LOCAL c,al$,f$,c%,c2%
  c=INP(2)
  '
  IF c>=187 AND c<=196
    exit=c-186
  ELSE
    INC nb_touche
    IF c=13 OR c=200 OR c=208 OR c=225 OR c=226 ! Ajouter un code SUITE
      IF buffer! AND nb_touche<=2 ! même s'il faut l'enlever après
        BYTE{buffer%+no%}=&H13
        BYTE{buffer%+SUCC(no%)}=&H48
        ADD no%,2
      ENDIF
      CLR nb_touche
    ENDIF
    SELECT c
    CASE 6   !Ctrl F
      OUT 1,&H1B,&H39,&H67
    CASE 159
      r%(4)=NOT r%(4)
    CASE 13  ! ENVOI
      OUT 1,&H13,&H41
    CASE 200 ! RETOUR
      OUT 1,&H13,&H42
    CASE 18  ! REPETITION
      OUT 1,&H13,&H43
    CASE 226 ! GUIDE
      OUT 1,&H13,&H44
    CASE 127 ! ANULATION
      OUT 1,&H13,&H45
    CASE 225 ! SOMMAIRE
      OUT 1,&H13,&H46
    CASE 8   ! CORRECTION
      OUT 1,&H13,&H47
    CASE 208 ! SUITE
```

```
      OUT 1,&H13,&H48
    CASE 3   ! CONNEXION FIN
      OUT 1,&H13,&H49
    CASE 19  ! Sauvegarde
      IF buffer!
        FILESELECT path$+"\*.min","",f$
        path$=LEFT$(f$,RINSTR(f$,"\")-1)
        IF f$<>""
          IF INSTR(f$,".")=0
            f$=f$+".min"
          ENDIF
          c%=buffer%
          REPEAT
            c%=C:instr2%(L:c%+sensible+2,
L:buffer%+no%)
            IF c%>0
              c2%=C:instr%(L:c%,L:buffer%+no%)
              IF c2%-c%<=sensible AND c2%>0
                SUB no%,2
                BMOVE c%,c%-2,buffer%+no%-c%
              ENDIF
            ENDIF
          UNTIL c%=0
          BSAVE f$,buffer%,PRED(no%)
          libere
          ALERT 1," Buffer effacé ",1,"OK",b!
        ENDIF
      ELSE
        ALERT 2,"Pas de buffer à sauver !!!",
1,"Ah ?",b!
      ENDIF
    CASE 94
      c=INP(2)
      IF c=97 OR c=101 OR c=105 OR c=111 OR c=117
        OUT 1,&H19,&H43,c
      ELSE
        OUT 1,94,c
      ENDIF
    CASE 185
      c=INP(2)
      OUT 1,&H19,&H48,c
    DEFAULT
      touch2(c)
    ENDSELECT
  ENDIF
RETURN
PROCEDURE touch2(c)
  SELECT c
  CASE 153,154,148,129,132,137,139,142
    OUT 1,&H19,&H49
    SELECT c
    CASE 129,154
      OUT 1,117
    CASE 132,142
      OUT 1,97
    CASE 137
      OUT 1,101
    CASE 139
      OUT 1,105
    CASE 148,153
      OUT 1,111
    ENDSELECT
  CASE 131,136,140,147,150
    OUT 1,&H19,&H43
    SELECT c
    CASE 131
      OUT 1,97
    CASE 136
      OUT 1,101
    CASE 140
      OUT 1,105
    CASE 147
      OUT 1,111
    CASE 150
      OUT 1,117
    ENDSELECT
  CASE 128,135
    OUT 1,&H16,75
    IF c=128
      OUT 1,67
    ELSE IF c=135
      OUT 1,99
    ENDIF
  CASE 130,144
    OUT 1,&H19,&H42,101
  CASE 138
    OUT 1,&H19,&H41,101
  CASE 182,133
    OUT 1,&H19,&H41,97
  CASE 151
    OUT 1,&H19,&H41,117
  CASE 248
    OUT 1,&H16,48
  CASE 221
    OUT 1,&H16,39
  CASE 156
    OUT 1,&H16,35
  DEFAULT
    OUT 1,c
  ENDSELECT
RETURN
PROCEDURE update_page
  no%=page%(page)
  PRINT AT(34,1);SUCC(page);"/";nb_pages;" ";
  efface=sensible
RETURN
PROCEDURE selectfich(titre$,chemin$,entree$,VAR nom$)
 FILESELECT #titre$,path$+chemin$,entree$,nom$
  IF nom$<>""
    path$=LEFT$(nom$,RINSTR(nom$,"\")-1)
  ENDIF
RETURN
`
` Fin
`
PROCEDURE fin
  libere
  ~XBIOS(5,L:-1,L:-1,resol)
  free_rs232
  ~MFREE(phys_alloc%)
  ~XBIOS(5,L:physbase%,L:physbase%,-1)
  RESERVE
  EDIT
RETURN
PROCEDURE libere
  IF buffer!
    VOID MFREE(buffer%)
    buffer!=FALSE
  ENDIF
  IF (NOT minitel!)
    result=MFREE(ad%)
  ENDIF
RETURN
`
PROCEDURE ressource ! Gère les paramètres
  LOCAL loc%,a$,x,loc2%
  `
  CLR no%
  SGET a$
  libere
` Libère toute la mémoire
  REPEAT
    CLS
    PRINT "Source :"
    PRINT "1 - Minitel"
    PRINT "2 - Fichier"
    INPUT "Choix ";x
    IF x=1
      minitel!=TRUE
      IF buff>=2 AND buff<300
        buffer!=TRUE
        long_buffer%=buff*1024
        buffer%=MALLOC(long_buffer%)
        IF buffer%<=0
          ALERT 3,"Quelque chose cloche :|
impossible d'allouer la mémoire|
nécessaire...",1,"Ah bon?",b
          fin
        ENDIF
      ENDIF
    ELSE IF x=2
      minitel!=FALSE
      selectfich("Fichier minitel","\*.MIN","",n$)
      IF EXIST(n$)
        OUT 1,&H1B,&H39,&H67
        OPEN "i",#1,n$
        long%=LOF(#1)
        ad%=MALLOC(long%)
        IF ad%>0
          BGET #1,ad%,long%
          CLOSE
        ELSE
          ALERT 3,"Quelque chose cloche :|
impossible d'allouer la mémoire|
nécessaire au chargement !",1,"OK",b
          fin
        ENDIF
        `
        ` Localisation des pages dans le fichier
        `
        CLR nb_pages,page
        loc%=ad%
        page%(0)=0
        INC nb_pages
        loc2%=loc%
        REPEAT
          IF loc2%>0
            loc2%=C:instr2%(L:loc%+sensible,
L:ad%+long%-1)
            ' fin page : SUITE
          ENDIF
          loc%=C:instr%(L:loc%+sensible,
L:ad%+long%-1)
          CLS
          IF loc%>0
            IF loc2%<loc% AND loc2%>0
              loc%=loc2%
            ENDIF
            page%(nb_pages)=loc%-ad%
            INC nb_pages
          ELSE IF loc2%>0
            loc%=loc2%
            page%(nb_pages)=loc%-ad%
            INC nb_pages
          ENDIF
        UNTIL loc%=0
      ELSE
        CLR x
      ENDIF
    ENDIF
  UNTIL (x=1 OR x=2)
  efface=sensible
  SPUT a$
RETURN
PROCEDURE alloc_rs232
  ` Allocation d'un tampon pour la RS
```

```
  LOCAL ad%
  rs232%=MALLOC(taille_rs%)
  ad%=XBIOS(14,0)
  old_rs%=LPEEK(ad%)
  LPOKE ad%,rs232%
  old_ta%=DPEEK(ad%+4)
  DPOKE ad%+4,taille_rs%
  LPOKE ad%+6,0
RETURN
PROCEDURE free_rs232
  ' Remet le tampon RS232 comme on l'a trouvé
```

```
  LOCAL ad%,err
  ad%=XBIOS(14,0)
  LPOKE ad%,old_rs%
  DPOKE ad%+4,old_ta%
  LPOKE ad%+6,0
  er=MFREE(rs232%)
  IF er
    ~FORM_ALERT(1,ERR$(er))
  ENDIF
RETURN
```

Saisissez le listing Assembleur 1, et, à la compilation (assemblez en mode PC relatif), sauvez sous le nom "MINI.PRG". Tapez le listing GfA 2, sauvegardez et lancez. Vous obtiendrez un fichier "MINI.INL", que vous intégrerez au listing GfA 1, dans le premier "Inline". Tapez les 2 autres listings Assembleur (toujours en mode PC relatif), à intégrer aussi dans les 2 autres Inline. Bonne saisie!
Si vous avez des problèmes, 3615 STMAG, rubrique *RTC.

LISTING GFA 2

```
CLS
SETCOLOR 0,0
SETCOLOR 15,&H777
PRINT AT(4,12);"Mémorisation de la fonte système"
PRINT AT(14,13);"Patientez..."
a$=STRING$(10000,CHR$(0))
IF NOT EXIST("A:\MINI.PRG")
  CLS
  PRINT "Routine assembleur introuvable !!!"
  PRINT "Interruption."
ELSE
  BLOAD "A:\MINI.PRG",V:a$
  deb=INSTR(300,a$,STRING$(40,CHR$(0)))+40+22+40*25*2+4*3
  FOR char=0 TO 95
    IF char>63
      car=char-32
    ELSE
      car=char
    ENDIF
    p=(car AND 1)*&HF0+(car AND 2)/2*&HF
    FOR t=0 TO 2
      POKE XBIOS(2)+char+t*160,p
    NEXT t
    FOR t=3 TO 5
      POKE XBIOS(2)+char+t*160,(car AND 4)/4*&HF0+(car AND 8)/8*&HF
    NEXT t
    FOR t=6 TO 7
      POKE XBIOS(2)+char+t*160,(car AND 16)/16*&HF0+(car AND 32)/32*&HF
    NEXT t
    car=char+32
    IF char<40
      dar=char
    ELSE
      dar=39
    ENDIF
    PRINT AT(1+dar,2);" ";
    SELECT car
    CASE 94
      LINE dar*8+3,8,dar*8+3,15
      DRAW dar*8+1,10 TO dar*8+3,8 TO dar*8+5,10
    CASE 96
      LINE dar*8,11,dar*8+7,11
    CASE 95
      LINE dar*8,15,dar*8+7,15
    CASE 123
      LINE dar*8,8,dar*8,15
    CASE 124
      LINE dar*8+3,8,dar*8+3,15
    CASE 125
      LINE dar*8+7,8,dar*8+7,15
    CASE 126
      LINE dar*8,8,dar*8+7,8
    CASE 127
      PBOX dar*8,8,dar*8+7,15
    CASE 92
      LINE dar*8,8,dar*8+7,15
    CASE 47
      LINE dar*8+7,8,dar*8,15
    DEFAULT
      PRINT AT(1+dar,2);CHR$(car);
    ENDSELECT
    IF MOD(dar,2)>0
      dar=SUCC(PRED(dar)*4)
    ELSE
      dar=dar*4
    ENDIF
    FOR t=0 TO 7
      POKE
V:a$+deb+768+t+char*8,PEEK(XBIOS(2)+
dar+t*160+1280)
      POKE
V:a$+deb+t+char*8,PEEK(XBIOS(2)+ch
ar+t*160)
    NEXT t
  NEXT char
  CLS
  PRINT AT(14,12);"Extensions..."
  DRAW 3,1 TO 0,3 TO 3,5
  LINE 0,3,7,3
  extend(1536) ! Flèche gauche (128)
  DRAW 0,3 TO 7,3 TO 4,1
  LINE 7,3,4,5
  extend(1544) ! Flèche droite (129)
  LINE 3,0,3,7
  DRAW 1,5 TO 3,7 TO 5,5
  extend(1552) ! Flèche basse (130)
  LINE 3,0,3,7
  DRAW 1,2 TO 3,0 TO 5,2
  extend(1560) ! Flèche haute (131)
  PRINT AT(1,1);"£";
  extend(1568) ! 132
  PRINT AT(1,1);"°";
  extend(1576) ! 133
  PRINT AT(1,1);"à";
  extend(1584) ! 134
  PRINT AT(1,1);"è";
  extend(1592) ! 135
  PRINT AT(1,1);"ù";
  extend(1600) ! 136
  PRINT AT(1,1);"é";
  extend(1608) ! 137
  PRINT AT(1,1);"â";
  extend(1616) ! 138
  PRINT AT(1,1);"ê";
  extend(1624) ! 139
  PRINT AT(1,1);"ô";
  extend(1632) ! 140
  PRINT AT(1,1);"î";
  extend(1640) ! 141
  PRINT AT(1,1);"û";
  extend(1648) ! 142
  PRINT AT(1,1);"ë";
  extend(1656) ! 143
  PRINT AT(1,1);"ä";
  extend(1664) ! 144
  PRINT AT(1,1);"ï";
  extend(1672) ! 145
  PRINT AT(1,1);"ö";
  extend(1680) ! 146
  PRINT AT(1,1);"ü";
  extend(1688) ! 147
  PRINT AT(1,1);"Ç";
  extend(1696) ! 148
  PRINT AT(1,1);"ç";
  extend(1704) ! 149
  PRINT AT(1,1);"±";
  extend(1712) ! 150
  PRINT AT(1,1);CHR$(246);
  extend(1720) ! 151
  PRINT AT(1,1);CHR$(172);
  extend(1728) ! 152
  PRINT AT(1,1);CHR$(171);
  extend(1736) ! 153
  PRINT AT(1,1);"Œ";
  extend(1744) ! 154
  PRINT AT(1,1);"œ";
  extend(1752) ! 155
  PRINT AT(1,1);"ß";
  extend(1760) ! 156
  PRINT AT(1,1);"§";
  extend(1768) ! 157
  CLS
  PRINT "Longueur totale de la
routine : ";deb+1776
  BSAVE A:\MINI.INL",V:a$,deb+1776
ENDIF
PROCEDURE extend(n)
  FOR t=0 TO 7
    POKE V:a$+deb+n+t,PEEK(XBIOS(2)+t*160)
    BYTE{XBIOS(2)+t*160}=0
  NEXT t
RETURN
```

LISTING ASSEMBLEUR 1

```
;* Routines d'émulation vidéotex
;* Version 3.20, 12 Avril 1990

nb_elements          = 2

 bra.s   init
 ; Branchements directs
 ; pour le GfA Basic.
 bra.s   main
 bra     newx
 bra     curseur

init:
 lea.l   phys,a0
 move.l  a0,d0
fin_init:
 lea.l   attrib,a5
 move.b  #$80,(a5)
 lea.l   encre,a5
 move.b  #7,(a5)
 rts

; * Programme principal: traite les
; * codes 8,9,10,11,12,13,14,15,
; * 17,20 et 24, ainsi que certains
; * codes ESCape.
; * X dans d0, Y dans d1, CAR dans d2,
; * FONTE dans d3, et ROULEAU! dans d4.
; * d5: boucles, d6 répétitions, et d7
; curseur.
;
; * d3:
; * bit 0 = Mode texte / mode graphique
; * bit 1 = Soulignement, texte,
;          masquage, graphique...
; * bit 2 = Double largeur
; * bit 3 = Double hauteur
; * bit 4 = Inversion vidéo.
; * a0 pointe sur l'adresse
;      écran concernée
; * a1 sur le caractère de la fonte
; * a2 sur un masque pour la
;      fonte graphique quadrillée
; * a3 sur la matrice écran.
; * a4 sur le fond (pas toujours...)
; * a5 sur l'attribut

main:
 cmpi.w  #31,d2
 ble     traite

 ; * Positionnement des pointeurs
 ; * de fontes

 bsr     convert
 lea.l   fond,a4
 lea.l   attrib,a5
 subi.w  #32,d2
 mulu.w  #8,d2
 ; D2 pointe sur le premier
 ; octet du caractère
 btst    #0,d3
 ; Serait-on en mode graphique?
 bne.s   graphic
 lea.l   768+fontes,a1
 ; Fonte normale: 768 octets plus loin
 bra.s   normal
graphic:
 lea.l   fontes,a1
 ; Fonte graphique: la première
 lea.l   masque,a2
 ; Si oui on prend le masque,
 ; ça peut servir...
normal:
 adda.l  d2,a1
 movem.w regs,d0-d2

; * ROUTINE PRINCIPALE

 btst    #3,d3
 ; Double hauteur
 beq.s   taille1
 cmpi.w  #2,d1
 bhi.s   dbletail
 bclr    #3,d3
 bclr    #2,d3
 lea.l   regs,a6
 move.w  d3,6(a6)
 bra.s   taille1
dbletail:
 move.w  d7,-(sp)
 move.w  #1,d7
bclh:
 suba.l  #40*4*8,a0
 suba.l  #40*nb_elements,a3
 lea.l   encre,a6
 ; Sert à indiquer à l'attribut
 ; de soulignement...
 bset    #5,(a6)
 ; ...que c'est le haut du caractère
appel:
 bsr     repet2
 lea.l   encre,a6
 bclr    #5,(a6)
 bsr     distance
 adda.l  #40*nb_elements,a3
 movem.w regs,d0-d4
 dbra    d7,appel
 move.w  (sp)+,d7
 cmpi.w  #32,d2
 bne.s   traisui
 bset    #7,(a5)
 bra.s   traisui
taille1:
 bsr.s   repet
 bsr     distance
traisui:
 bsr     fin_aff
 dbra    d6,main
 move.w  #0,d6
fin_gen:
 tst.w   d7
 ; Si d7<>0 alors afficher curseur
 bne     curseur
 rts

 ; * BOUCLE PRINCIPALE

repet:
 btst    #0,d3
 beq.s   textr
 move.b  attrib,(a4)
 bset    #7,(a5)
 bclr    #7,(a4)
 cmpi.w  #1,d0
 beq.s   repdeb
 btst    #6,-1(a3)
 ; Le soulignement du
 ; carac d'avant!
 beq.s   effsoul
 bset    #6,(a4)
 bra.s   repdeb
effsoul:
 bclr    #6,(a4)
 bra.s   repdeb

textr:
 cmpi.b  #32,d2
 bne.s   sans_att
 btst    #7,(a5)
 bne.s   sans_att
 move.b  (a5),(a4)
 ; Attrib -> fond
 bset    #7,(a5)
 bset    #5,(a4)
 bra.s   repdeb
sans_att:
 cmpi.w  #1,d0
 bhi.s   attrn
 move.b  #$80,(a4)
 bra.s   repdeb
attrn:
 move.b  -1(a3),(a4)
 bset    #7,(a4)
 bclr    #5,(a4)
repdeb:
 move.b  fond,1(a3)
 move.b  encre,(a3)
 btst    #1,d3
 ; Si pas de masquage...
 beq     aff_ok
 ; ...on va voir ailleurs

; * Quand le programme arrive ici,
; * c'est qu'on est en masquage de
; * caractères graphiques...

 move.w  #7,d5
bouc_msk:
 move.b  (a1)+,d2
 and.b   (a2)+,d2
 bsr     affich
 adda.l  #160,a0
 dbra    d5,bouc_msk
 rts

 ; * Cas où il n'y a pas d'inversion
 ; * vidéo ni de masquage...

aff_ok:
 btst    #0,d3
 ; Est-on en mode graphique?
 bne     text1
 ; Si oui il y a une seule
 ; taille possible
 ; * Le mode graphique est maintenant
 ; * complètement traité.
 ; * Reste 3 possibilités:
 ; * - double hauteur
 ; * - double largeur
 ; * - standard

aff_ok2:
 btst    #2,d3
 beq.s   sui_test
 cmpi.w  #40,d0
 bne     dble_lar
sui_test:
 btst    #4,d3
 ; Inverse vidéo
 bne     text1i

; * Routine d'affichage la plus
; * standard qui soit. Vérifie
```

```
; * quand même le soulignement...
text1:
 move.w  #7,d5
bouc_nor:
 move.b  (a1)+,d2
 bsr     affich
 adda.l  #160,a0
 dbra    d5,bouc_nor
 cmpi.w  #1,d0
 beq.s   fin_nor
 btst    #6,-1(a3)
 ; Soulignement?
 beq.s   fin_nor
 suba.l  #160,a0
 bsr     soulign1
 adda.l  #160,a0
fin_nor:
 tst.w   d6
 beq.s   fin_nor2
 suba.l  #8,a1
 bsr     fin_aff
 cmpi.w  #1,d0
 beq.s   repar2
 lea.l   regs,a4
 movem.w d0/d1,(a4)
 suba.l  #1279,a0
 move.w  a0,d0
 btst    #0,d0
 bne.s   t1nor
 adda.l  #6,a0
t1nor:
 adda.l  #nb_elements,a3
 move.b  fond,1(a3)
 move.b  encre,(a3)
 dbra    d6,text1
repar2:
 bsr     convert
 move.b  fond,1(a3)
 move.b  encre,(a3)
 dbra    d6,text1
fin_nor2:
 rts

; * On reprend la même, sauf qu'on
; *  fait de l'inversion vidéo
text1i:
 bset    #6,(a3)
 ; Marque de l'inversion vidéo
 ; pour l'attribut à distance
 move.w  #7,d5
bouc_nr2:
 move.b  (a1)+,d2
 not.b   d2
 bsr     affich
 adda.l  #160,a0
 dbra    d5,bouc_nr2
 cmpi.w  #1,d0
 beq.s   fin_nr2
 btst    #6,-1(a3)
 ; Soulignement?
 beq     fin_nr2
 suba.l  #160,a0
 bsr     soulign2
 adda.l  #160,a0
fin_nr2:
 rts

; * Il ne reste plus que la
; *  double largeur...
dble_lar:
 move.b  fond,(nb_elements+1)(a3)
 move.b  encre,(nb_elements)(a3)
```

```
 btst    #4,d3
 ; Inversion vidéo
 bne     text3i

; * Traitement double largeur

lar_stnd:
 move.w  d6,-(sp)
 move.w  #7,d6

sui_larg:
 move.w  #7,d0
 move.b  (a1),d1
 ; d1=d5 et a1 pointe sur
 ; l'octet à traiter
 move.b  (a1)+,d5
bouc_lar:
 roxr.b  #1,d1
 roxr.w  #1,d2
 roxr.b  #1,d5
 roxr.w  #1,d2
 ; D2 CONTIENT 2 FOIS
 ; LE MêME BIT
 dbra    d0,bouc_lar

 ; à ce stade, d2 contient
 ; l'octet "grossi"

 move.w  a0,d0
 btst    #0,d0
 bne.s   non_st
 addq.l  #1,a0
 bsr     affich
 subq.l  #1,a0
 bra.s   aff_lar2
non_st:
 addq.l  #7,a0
 bsr     affich
 subq.l  #7,a0
aff_lar2:
 ror.w   #8,d2
 bsr     affich
 ; (on ne peut pas traiter
 ; le mot globalement)
 adda.l  #160,a0
 dbra    d6,sui_larg
 lea.l   regs,a2
 cmpi.w  #1,(a2)
 beq.s   fin_lar1
 btst    #6,-1(a3)
 ; Soulignement?
 beq     fin_lar1
 suba.l  #160,a0
 bsr     soulign1
 addq.l  #1,a0
 move.w  a0,d0
 btst    #0,d0
 beq.s   larsoul2
 bsr     soulign1
 bra.s   fin_lar1
larsoul2:
 addq.l  #6,a0
 bsr     soulign1
fin_lar1:
 move.w  (sp)+,d6
 rts

; * Le plus dur est fait.
; * Maintenant inversion vidéo!

text3i:
```

```
 bset    #6,(a3)
 ; Marquage inversion vidéo
 bset    #6,(nb_elements)(a3)
 move.w  d6,-(sp)
 move.w  #7,d6
sui_lar2:
 move.w  #7,d0
 move.b  (a1),d1
 ; d1=d2 et a1 pointe sur
 ; l'octet à traiter
 move.b  (a1)+,d5
bouc_lar2:
 roxr.b  #1,d1
 roxr.w  #1,d2
 roxr.b  #1,d5
 roxr.w  #1,d2
 ; D2 CONTIENT 2 FOIS
 ; LE MêME BIT
 dbra    d0,bouc_lar2

 ; à ce stade, d5 contient
 ; l'octet "grossi"

 not.w   d2
 move.w  a0,d0
 btst    #0,d0
 bne.s   non_st2
 addq.l  #1,a0
 bsr     affich
 subq.l  #1,a0
 bra.s   aff_la2b
non_st2:
 addq.l  #7,a0
 bsr     affich
 subq.l  #7,a0
aff_la2b:
 ror.w   #8,d2
 bsr     affich
 ; (on ne peut pas traiter
 ; le mot globalement)
 adda.l  #160,a0
 dbra    d6,sui_lar2
 lea.l   regs,a2
 cmpi.w  #1,(a2)
 beq.s   fin_lar2
 btst    #6,-1(a3)
 ; Soulignement?
 beq     fin_lar2
 suba.l  #160,a0
 bsr     soulign2
 addq.l  #1,a0
 move.w  a0,d0
 btst    #0,d0
 beq.s   larsoul3
 bsr     soulign2
 bra.s   fin_lar2
larsoul3:
 addq.l  #6,a0
 bsr     soulign2
fin_lar2:
 move.w  (sp)+,d6
 rts

;* TRAITEMENT DE LA DOUBLE HAUTEUR

repet2:
 cmpi.w  #32,d2
 bne.s   sans_ath
 btst    #7,(a5)
 bne.s   sans_ath
 move.b  (a5),(a4)
 ; Attrib -> fond
```

```
 bset    #5,(a4)
 ; Un attribut, un vrai!
 bra.s   repdeb2
sans_ath:
 cmpi.w  #1,d0
 bhi.s   attrn2
 move.b  #$80,(a4)
 bra.s   repdeb2
attrn2:
 move.b  -1(a3),(a4)
 bset    #7,(a4)
 ; Ce n'est pas un attribut
 bclr    #5,(a4)
 ; Ah mais vraiment pas alors
repdeb2:
 move.b  fond,1(a3)
 move.b  encre,(a3)

; * Cas où il n'y a pas d'inversion
; * vidéo ni de masquage...
aff2_ok2:
 btst    #2,d3
 beq.s   tests2
 cmpi.w  #40,d0
 bne     dble2_lar
tests2:
 btst    #4,d3
 ; Inverse vidéo
 bne     text1i2

; * Routine d'affichage la plus
; * standard qui soit. Vérifie
; * quand même le soulignement...
text12:
 move.w  #3,d5
bouc2_nor:
 move.b  (a1)+,d2
 bsr     affich
 adda.l  #160,a0
 bsr     affich
 adda.l  #160,a0
 dbra    d5,bouc2_nor
 btst    #6,-1(a3)
 ; Soulignement?
 beq.s   finnor2
 btst    #5,(a3)
 bne.s   finnor2
 suba.l  #160,a0
 bsr     soulign1
 adda.l  #160,a0
finnor2:
 rts

; * On reprend la même, sauf
; * qu'on fait de l'inversion vidéo!
text1i2:
 move.w  #3,d5
bouc2_nr2:
 move.b  (a1)+,d2
 not.b   d2
 bsr     affich
 adda.l  #160,a0
 bsr     affich
 adda.l  #160,a0
 dbra    d5,bouc2_nr2
 btst    #6,-1(a3)
 ; Soulignement?
 beq.s   fini2
 btst    #5,(a3)
 bne.s   fini2
 suba.l  #160,a0
 bsr     soulign2
 adda.l  #160,a0
fini2:
 rts

; * Il ne reste plus que la
; * double largeur...
dble2_lar:
 move.b  fond,(1+nb_elements)(a3)
 move.b  encre,(nb_elements)(a3)

 btst    #4,d3
 ; Inversion vidéo
 bne     text3i2

lar2_stnd:
 movem.w d1/d6/d7,-(sp)
 move.w  #3,d6
hautlar:
 move.w  #1,d7

sui2_larg:
 move.w  #7,d0
 move.b  (a1),d1
 ; D1=D5 et a1 pointe sur
 ; l'octet à traiter
 move.b  d1,d5
bouc2_lar:
 roxr.b  #1,d1
 roxr.w  #1,d2
 roxr.b  #1,d5
 roxr.w  #1,d2
 ; D2 CONTIENT 2 FOIS
 ; LE MêME BIT
 dbra    d0,bouc2_lar
 ; à ce stade, d2 contient
 ; l'octet "grossi"
 move.w  a0,d0
 btst    #0,d0
 bne.s   non2_st
 addq.l  #1,a0
 bsr     affich
 subq.l  #1,a0
 bra.s   aff2_lar2
non2_st:
 addq.l  #7,a0
 bsr     affich
 subq.l  #7,a0
aff2_lar2:
 ror.w   #8,d2
 bsr     affich
 ; (on ne peut pas traiter
 ; le mot globalement)
 adda.l  #160,a0
 dbra    d7,sui2_larg
 addq.l  #1,a1
 dbra    d6,hautlar

 btst    #6,-1(a3)
 ; Soulignement?
 beq.s   fin2_lar1
 btst    #5,(a3)
 bne.s   fin2_lar1
 suba.l  #160,a0
 bsr     soulign1
 move.w  a0,d0
 btst    #0,d0
 bne.s   lar2soul2
 addq.l  #1,a0
 bsr     soulign1
 bra.s   fin2soul2
lar2soul2:
 addq.l  #7,a0
 bsr     soulign1
fin2soul2:
 adda.l  #160,a0
fin2_lar1:
 movem.w (sp)+,d1/d6/d7
 rts

; * Double largeur
; * inverse vidéo
text3i2:
 movem.w d1/d6/d7,-(sp)
 move.w  #3,d6
hautlari:
 move.w  #1,d7
sui2_lar2:
 move.w  #7,d0
 move.b  (a1),d1
 ; d1=d2 et a1 pointe sur
 ; l'octet à traiter
 move.b  d1,d5
bouc2_lar2:
 roxr.b  #1,d1
 roxr.w  #1,d2
 roxr.b  #1,d5
 roxr.w  #1,d2
 ; D2 CONTIENT 2 FOIS
 ; LE MêME BIT
 dbra    d0,bouc2_lar2
 ; à ce stade, d5 contient
 ; l'octet "grossi"
 not.w   d2
 move.w  a0,d0
 btst    #0,d0
 bne.s   non2_st2
 addq.l  #1,a0
 bsr     affich
 subq.l  #1,a0
 bra.s   aff2_la2b
non2_st2:
 addq.l  #7,a0
 bsr     affich
 subq.l  #7,a0
aff2_la2b:
 ror.w   #8,d2
 bsr     affich
 ; (on ne peut pas traiter
 ; le mot globalement)
 adda.l  #160,a0
 dbra    d7,sui2_lar2
 addq.l  #1,a1
 dbra    d6,hautlari
 btst    #6,-1(a3)
 ; Soulignement?
 beq.s   fin2_lar2
 btst    #5,(a3)
 bne.s   fin2_lar2
 suba.l  #160,a0
 bsr     soulign2
 btst    #0,d0
 bne     lar2soul3
 addq.l  #1,a0
 bsr     soulign2
 bra.s   fin2soul2
lar2soul3:
 addq.l  #7,a0
 bsr     soulign2
 adda.l  #160,a0
fin2_lar2:
 movem.w (sp)+,d1/d6/d7
 rts

; * C'est fini!
```

```
fin_aff:
 movem.w regs,d0-d4
 ; On récupère les valeurs
 ; x/y et ASCII
 bsr.s   newx
 btst    #2,d3
 beq.s   fintot
 cmpi.w  #1,d0
 bne.s   newx
fintot:
 rts

; * Procédures

; Allonge l'attribut
; si besoin est

distance:
 move.w  regs,d0
 movem.l a0/a3/a4,-(sp)
 move.b  fond,d4
 lea.l   encre,a4
 move.b  (a4),d5
 andi.b  #$f,d4
 btst    #2,d3
 beq.s   disnorm
 addq.l  #nb_elements,a3
 addq.b  #1,d0
 addq.l  #1,a0
 move.w  a0,d1
 btst    #0,d1
 bne.s   disnorm
 addq.l  #6,a0
disnorm:
 addq.l  #nb_elements,a3
 cmpi.w  #40,d0
 ; Si fin de ligne
 bge.s   fin
 move.b  1(a3),d1
 move.b  fond,d2
 bclr    #7,d2
 bclr    #7,d1
 cmp.b   d2,d1
 beq.s   fin
 btst    #7,1(a3)
 ; Si autre attribut
 beq.s   fin
 andi.b  #$f0,1(a3)
 or.b    d4,1(a3)
 move.b  (a3),(a4)
 addq.b  #1,d0
 suba.l  #40*4*8-1,a0
 move.w  a0,d1
 btst    #0,d1
 bne.s   incnorm
 addq.l  #6,a0
incnorm:
 move.w  #7,d1
disbouc:
 move.b  6(a0),d2
 bsr     affich
 lea.l   160(a0),a0
 dbra    d1,disbouc

; * Ce cher soulignement...
 btst    #6,fond
 ; Soulignement
 beq.s   disnorm
 bset    #6,1(a3)
 ; Soulignement dans la matrice
 btst    #5,(a3)
 ; Haut d'un caractère double hauteur
 bne.s   disnorm
 lea.l   -160(a0),a0
 btst    #6,(a3)
 ; Inversion vidéo
 bne.s   sinvdis
 ; CAD: Soulignement INVersé
 ; et à DIStance!
 move.b  (a3),d1
 addq.l  #4,a0
 move.w  #2,d3
dbsoul1:
 btst    d3,d1
 beq.s   dsvid1
 move.b  #$ff,(a0)
 subq.l  #2,a0
 dbra    d3,dbsoul1
 bra.s   dfsoul1
dsvid1:
 move.b  #0,(a0)
 subq.l  #2,a0
 dbra    d3,dbsoul1
dfsoul1:
 lea.l   162(a0),a0
 bra.s   disnorm

sinvdis:
 move.b  1(a3),d1
 addq.l  #4,a0
 move.w  #2,d3
 bra.s   dbsoul1

fin:
 move.b  d5,(a4)
 movem.l (sp)+,a0/a3/a4
 rts

; * Nouvelles coordonnées
; * du curseur après
; * affichage d'un carac

newx:
 addq.w  #1,d0
 ; Renouvellement des x,y
 cmpi.w  #40,d0
 ble     fin_x
 cmpi.w  #1,d1
 bne     suit_aff
 move.w  #40,d0
 bra.s   fin_x
suit_aff:
 addq.w  #1,d1
 subi.w  #40,d0
 cmpi.w  #25,d1
 ble.s   fin_y
 tst.w   d4
 bne.s   roule
 move.w  #2,d1
 bra.s   fin_y
roule:
 move.w  #25,d1
 bsr     scrollh
fin_y:
 btst    #3,d3
 ; Double hauteur?
 beq.s   fin_x
 addq.w  #1,d1
 cmpi.w  #25,d1
 ble     fin_x
 tst.w   d4
 bne.s   roule2
 move.w  #2,d1
 bra.s   fin_x
roule2:
 move.w  #25,d1
 bra.s   scrollh
fin_x:
 rts

; * Routine d'affichage couleur
affich:
 movem.w d3-d5,-(sp)
 move.b  encre,d4
 move.b  fond,d3
 addq.l  #4,a0
 move.b  d2,2(a0)
 move.w  #2,d5
affb1:
 btst    d5,d3
 beq.s   affvid1
 move.b  #255,(a0)
 btst    d5,d4
 bne.s   affin
 and.b   d2,(a0)
 not.b   (a0)
 bra.s   affin
affvid1:
 move.b  #0,(a0)
 btst    d5,d4
 beq.s   affin
 move.b  d2,(a0)
affin:
 subq.l  #2,a0
 dbra    d5,affb1
 addq.l  #2,a0
 movem.w (sp)+,d3-d5
 rts

soulign1:
 btst    #7,1(a3)
 ; Attribut?
 beq.s   fin_x
 ; Et bin alors pas
 ; de soulignement
soulok:
 movem.w d4/d5,-(sp)
 move.b  encre,d4
 addq.l  #4,a0
 move.w  #2,d5
bsoul1:
 btst    d5,d4
 beq.s   svid1
 move.b  #$ff,(a0)
 suba.l  #2,a0
 dbra    d5,bsoul1
 bra.s   finsoul1
svid1:
 move.b  #0,(a0)
 suba.l  #2,a0
 dbra    d5,bsoul1
finsoul1:
 addq.l  #2,a0
 movem.w (sp)+,d4/d5
 rts

soulign2:
 btst    #7,1(a3)
 beq     fin_x
sou2ok:
 movem.w d4/d5,-(sp)
 move.b  fond,d4
 addq.l  #4,a0
 move.w  #2,d5
bsoul2:
 btst    d5,d4
```

```
 beq.s    svid2
 move.b   #$ff,(a0)
 suba.l   #2,a0
 dbra     d5,bsoul2
 bra.s    finsoul1
svid2:
 move.b   #$00,(a0)
 suba.l   #2,a0
 dbra     d5,bsoul2
 addq.l   #2,a0
 movem.w  (sp)+,d4/d5
 rts

; * Routine de traitement des
; * codes de contrôle
traite:
 tst.w    d7
 beq.s    gauche
 bsr      curseur

;* Traitement des codes de
;* contrôle Vidéotex
gauche:
 ; Traitements des codes <32
 cmpi.w   #8,d2
 bne.s    droite
 cmpi.w   #1,d1
 beq      fin_gen
 subq.w   #1,d0
 bne      fin_gen
 move.w   #40,d0
 subq.w   #1,d1
 cmpi.w   #1,d1
 bne      fin_gen
 tst.w    d4
 beq.s    roulgau
 move.w   #2,d1
 bsr      scrolld
 bra.s    fin_gen
roulgau:
 move.w   #25,d1
 bra.s    fin_gen

droite:
 cmpi.w   #9,d2
 bne.s    bastst
 move.w   d3,-(sp)
 move.w   #0,d3
 bsr      newx
 move.w   (sp)+,d3
 bra.s    fin_gen

bastst:
 cmpi.w   #10,d2
 bne      haut
 cmpi.w   #1,d1
 bne      bas
 move.b   coord,d0
 move.b   coord+1,d1
 lea.l    encre,a4
 move.b   color,(a4)
 move.b   memd3,d3
 lea.l    attrib,a4
 move.b   mematt,(a4)
 bra.s    fin_gen
bas:
 addq.w   #1,d1
 cmpi.w   #26,d1
 bne      fin_gen
 tst.w    d4
 beq      bas2
 move.w   #25,d1
 bsr      scrollh
 bra.s    fin_gen
bas2:
 move.w   #2,d1
 bra.s    fin_gen

haut:
 cmpi.w   #11,d2
 bne.s    cde24
 cmpi.w   #1,d1
 beq      fin_gen
 subq.w   #1,d1
 cmpi.w   #1,d1
 bne      fin_gen
 tst.w    d4
 beq      haut2
 move.w   #2,d1
 bsr      scrolld
 bra      fin_gen
haut2:
 move.w   #25,d1
 bra.s    fin_gen

; * Traitement du code 24

cde24:
 cmpi.w   #24,d2
 bne.s    ctrl

 ; Positionement de l'attribut de fond,
 ; qui sera le même pour toute la ligne

 bsr      convert
 lea.l    fond,a4
 move.w   regs,d0
 btst     #0,d3
 ; Mode graphique
 beq.s    norm24
 move.b   attrib,(a4)
 bclr     #7,(a4)
 bra.s    b24
norm24:
 cmpi.w   #1,d0
 bne.s    attrn24
 move.b   #$80,(a4)
 bra.s    b24
attrn24:
 move.b   -1(a3),(a4)
 bset     #7,(a4)
 bclr     #5,(a4)

b24:
 btst     #4,d3
 beq.s    vide24
 lea.l    encre,a2
 bset     #6,(a2)
 move.b   #$ff,d2
 bra.s    post24
vide24:
 move.b   #0,d2
post24:
 btst     #3,d3
 ; Double hauteur?
 beq.s    taill124

 ; code 24 double hauteur
 suba.l   #40*8*4,a0
 suba.l   #40*nb_elements,a3
 bsr.s    bouc24
 adda.l   #40*nb_elements,a3
taill124:
 bsr.s    bouc24
 addq.b   #1,d0
 addq.l   #nb_elements,a3
 move.w   a0,d1
 btst     #0,d1
 bne.s    impair24
 suba.l   #1279,a0
 cmpi.b   #41,d0
 bne.s    post24
 bra.s    stest24
impair24:
 suba.l   #1273,a0
 cmpi.b   #41,d0
 bne.s    post24

stest24:
 btst     #6,(a4)
 ; Soulignement?
 beq.s    suit24
 btst     #0,d3
 ; Mode graphique?
 bne.s    suit24
 movem.w  regs,d0/d1
 bsr      convert
 adda.l   #40*4*7,a0
 ; +7 lignes
 move.w   regs,d0
 btst     #4,d3
 bne.s    invsou24
sou24:
 bsr      soulign1
 adda.l   #1,a0
 move.w   a0,d1
 btst     #0,d1
 bne.s    nbouc24
 adda.l   #6,a0
nbouc24:
 addq.b   #1,d0
 cmpi.b   #41,d0
 bne.s    sou24
 bra.s    suit24
invsou24:
 bsr      soulign2
 adda.l   #1,a0
 move.w   a0,d1
 btst     #0,d1
 bne.s    ibouc24
 adda.l   #6,a0
ibouc24:
 addq.b   #1,d0
 cmpi.w   #41,d0
 bne.s    invsou24

suit24:
 movem.l  erase,d0/d1/d6
 movem.w  regs,d0-d4
 tst.w    d7
 bne.s    curseur
 rts

bouc24:
 move.w   #7,d1
 ; On se sert de D2 pour déterminer
 ; si on est en inversion vidéo
 tst.w    d2
 bne.s    inv24
cara24:
 move.b   #0,6(a0)
 bsr      sou2ok
 adda.l   #160,a0
 dbra     d1,cara24
 move.b   encre,(a3)
 move.b   (a4),1(a3)
```

```
 rts

inv24:
 move.b  #$ff,6(a0)
 bsr     soulok
 adda.l  #160,a0
 dbra    d1,inv24
 move.b  encre,(a3)
 move.b  (a4),1(a3)
 rts

ctrl:
 cmpi.w  #12,d2
 bne.s   debline
 lea.l   encre,a0
 move.b  #7,(a0)
 bsr     cls
 bra.s   fin_gen

debline:
 cmpi.w  #13,d2
 bne.s   curson
 move.w  #1,d0
 bra.s   fin_gen

curson:
 cmpi.w  #17,d2
 bne     cursoff
 move.w  #-1,d7
 bra.s   fin_gen

cursoff:
 cmpi.w  #20,d2
 bne.s   gra
 clr.w   d7
 rts

gra:
 cmpi.w  #14,d2
 bne.s   texte
 move.w  #1,d3
 lea.l   attrib,a6
 andi.b  #$3f,(a6)
 bra.s   fin_gen

texte:
 cmpi.w  #15,d2
 bne.s   home
 move.w  #0,d3
 lea.l   attrib,a6
 andi.b  #$bf,(a6)
 bra.s   fin_gen

home:
 cmpi.w  #30,d2
 bne     fin_gen
 move.w  #1,d0
 move.w  #2,d1
 move.w  #0,d3
 bsr     fin_init
 bra.s   fin_gen

; Bascule le curseur coordonnées
; écran: (d0,d1).
; CONVERT convertit les coordonnées
; écran en adresses.
curseur:
 movem.w d0/d1,-(sp)
 bsr     convert
 move.w  #7,d0
hide2:
 not.b   (a0)
 not.b   2(a0)
 not.b   (4)(a0)
 adda.l  #160,a0
 dbra    d0,hide2
 movem.w (sp)+,d0/d1
 rts

convert:
 lea.l   regs,a4
 movem.w d0-d4,(a4)
 subq.w  #1,d0
 ; Conversion x,y en
 ; adresse écran
 mulu.w  #nb_elements,d0
 subq.w  #1,d1
 lea.l   matrice,a3
 adda.l  d0,a3
 mulu.w  #nb_elements*40,d1
 adda.l  d1,a3
 movem.w regs,d0/d1
 subq.w  #1,d0
 subq.w  #1,d1
 btst    #0,d0
 bne.s   convmul2
 mulu.w  #4,d0
 bra.s   convsuit
convmul2:
 mulu.w  #4,d0
 subq.w  #3,d0
convsuit:
 mulu.w  #40*4*8,d1
 movea.l phys,a0
 adda.l  d0,a0
 adda.l  d1,a0
 rts

; Efface l'écran
; à partir de la deuxième ligne

cls:
 movem.l d0-d7,-(sp)
 lea.l   matrice,a3
 adda.l  #40*nb_elements*25,a3
 ; Fin de l'écran
 movem.l erase,d1-d7/a4-a6
 move.w  #23,d0
clsbc1:
 movem.l d1-d7/a4-a6,-(a3)
 movem.l d1-d7/a4-a6,-(a3)
 dbra    d0,clsbc1
 movem.l (sp)+,d0-d7
 movea.l phys,a0
 adda.l  #1280,a0
 move.w  #7679,d0
boucle:
 move.l  #0,(a0)+
 dbra    d0,boucle
 move.w  #1,d0
 move.w  #2,d1
 clr.w   d3
 lea.l   coord,a0
 move.l  #0,(a0)
 rts

; Scrolling vers le haut

scrollh:
 movem.l d0-d4/d6/d7/a1,-(sp)
 lea.l   matrice,a3
 adda.l  #40*nb_elements,a3
 ; 2ème ligne
 movea.l a3,a4
 adda.l  #40*nb_elements,a4
 ; 3ème ligne
 move.w  #23-1,d0
 ; 25-2 lignes

scrhbc:
 movem.l (a4)+,d1-d7/a0-a2
 movem.l d1-d7/a0-a2,(a3)
 movem.l (a4)+,d1-d7/a0-a2
 movem.l d1-d7/a0-a2,40(a3)
 lea.l   80(a3),a3
 dbra    d0,scrhbc

 movem.l erase,d1-d7/a4-a6
 movem.l d1-d7/a4-a6,(a3)
 movem.l d1-d7/a4-a6,40(a3)

; Copie la 1ère ligne 1 ligne
; plus bas

 movea.l phys,a0
 movea.l a0,a1
 adda.l  #$500,a1
 move.w  #3,d0
ligne1h:
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,48(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,96(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,144(a1)

 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,192(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,240(a1)
 movem.l (a0)+,d1-d7/a2
 movem.l d1-d7/a2,288(a1)
 lea.l   320(a1),a1
 dbra    d0,ligne1h

 movea.l phys,a0
 adda.l  #$500,a0
 cmpa.l  physmax,a0
 bne.s   deplh

 ; Déplace physiquement
 ; l'écran vers physmed si
 ; nécessaire

 movea.l physmed,a1
 move.w  #4*24-1,d0
scroll1:
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,48(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,96(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,144(a1)

 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,192(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,240(a1)
 movem.l (a0)+,d1-d7/a2
 movem.l d1-d7/a2,288(a1)

 lea.l   320(a1),a1
```

```
 dbra    d0,scroll1
 movea.l physmed,a0

deplh:
 lea.l   phys,a1
 move.l  a0,(a1)

 adda.l  #32000,a0
 movem.l erase,d1-d7/a3-a5
 move.w  #3,d0
ctrly:
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)
 movem.l d1-d7/a3-a5,-(a0)

 dbra    d0,ctrly
 move.w  #-1,-(sp)
 move.l  phys,-(sp)
 move.l  phys,-(sp)
 move.w  #5,-(sp)
 trap    #$e
 adda.l  #12,sp
 movem.l (sp)+,d0-d4/d6/d7/a1
 rts

 ; Scrolling vers le bas
 ; Concerne le code 11

scrolld:
 movem.l d0-d4/d6/d7,-(sp)
 lea.l   matrice,a3
 adda.l  #40*25*nb_elements,a3
 ; Fin d'écran
 movea.l a3,a4
 suba.l  #40*nb_elements+40,a4
 move.w  #23-1,d0

scrdbc:
 movem.l (a4),d1-d7/a0-a2
 movem.l d1-d7/a0-a2,-(a3)
 movem.l -40(a4),d1-d7/a0-a2
 movem.l d1-d7/a0-a2,-(a3)
 lea.l   -80(a4),a4
 dbra    d0,scrdbc

 lea.l   matrice,a3
 adda.l  #40*nb_elements,a3
 movem.l erase,d1-d7/a4-a6
 movem.l d1-d7/a4-a6,(a3)
 movem.l d1-d7/a4-a6,40(a3)

; Copie la 1ère ligne 1 ligne
; au-dessus

 movea.l phys,a0
 movea.l a0,a1
 suba.l  #$500,a1
 move.w  #3,d0
ligne2h:
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,48(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,96(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,144(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,192(a1)
 movem.l (a0)+,d1-d7/a2-a6
 movem.l d1-d7/a2-a6,240(a1)
 movem.l (a0)+,d1-d7/a2
 movem.l d1-d7/a2,288(a1)
 lea.l   320(a1),a1
 dbra    d0,ligne2h

 movea.l phys,a0
 suba.l  #$500,a0
 cmpa.l  physmin,a0
 bne.s   deplb

 ; Déplace physiquement
 ; l'écran si nécessaire

 adda.l  #32000-48,a0
 movea.l physmed,a1
 adda.l  #32000,a1
 move.w  #4*25-1,d0
scroll2:
 movem.l (a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -48(a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -96(a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -144(a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -192(a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -240(a0),d1-d7/a2-a6
 movem.l d1-d7/a2-a6,-(a1)
 movem.l -272(a0),d1-d7/a2
 movem.l d1-d7/a2,-(a1)
 lea.l   -320(a0),a0
 dbra    d0,scroll2
 movea.l physmed,a0

deplb:
 lea.l   phys,a1
 move.l  a0,(a1)
 movem.l erase,d1-d7/a3-a5
 move.w  #7,d0
 movea.l phys,a2
 adda.l  #2560,a2
ctrly2:
 movem.l d1-d7/a3-a5,-(a2)
 movem.l d1-d7/a3-a5,-(a2)
 movem.l d1-d7/a3-a5,-(a2)
 movem.l d1-d7/a3-a5,-(a2)
 dbra    d0,ctrly2
 move.w  #-1,-(sp)
 move.l  a0,-(sp)
 move.l  a0,-(sp)
 move.w  #5,-(sp)
 trap    #$e
 adda.l  #12,sp

 movem.l (sp)+,d0-d4/d6/d7
 rts

;
; Zone des données : il y a 2 fontes,
; une pour le texte, et l'autre pour le
; semi-graphique. Elles doivent être mises
; en place par l'utilisateur.
; Elles sont suivies des accents et autres
; caractères spéciaux.

        .DATA

masque: .DC.b 119,119,0,119,0,119,119,0
erase:  .DC.l 0,0,0,0,0,0,0,0,0,0

        .BSS
phys:   .DS.l 1
coord:  .DS.w 1
color:  .DS.b 1
attrib: .DS.b 1
fond:   .DS.b 1
encre:  .DS.b 1
memd3:  .DS.b 1
mematt: .DS.b 1

physmin: .DS.l 1
physmax: .DS.l 1
physmed: .DS.l 1

regs:   .DS.w 5
matrice: .DS.b 40*25*nb_elements
fontes: .DS.w 1
        .END
```

LISTING ASSEMBLEUR 2

```
; INSTR2.L
; Recherche du prochain SUITE
; Paramètres: à partir de, longueur max

 movea.l 4(sp),a0
 movea.l 8(sp),a1
boucle:
 cmpi.b  #$19,(a0)+
 bne.s   suite
 cmpi.b  #$48,(a0)+
 beq.s   fin
suite:
 cmpa.l  a0,a1
 bhi.s   boucle
 movea.l #0,a0
fin:
 move.l  a0,d0
 rts
```

LISTING ASSEMBLEUR 3

```
; INSTR.L

; Recherche du prochain CLS
; Paramètres: adresse de départ, longueur
maximale

 movea.l 4(sp),a0
 movea.l 8(sp),a1
boucle:
 cmpi.b  #12,(a0)+
 beq.s   fin
 cmpa.l  a0,a1
 bhi.s   boucle
 movea.l #1,a0
fin:
 move.l  a0,d0
 subq.l  #1,d0
 rts
```

LES PETITES ANNONCES

VENTES - RP

Vends 1040 STF, souris, moniteur couleur SC 1425 (11/89) + 50 disquettes, le tout 5400F. Tél : (1) 39 85 90 90 ou 24 54 21 20.

Vends Atari 1040 STF (lecteur double face), lecteur externe, moniteur mono SM 125, imprimante Citizen 120D, softs, le tout 5000F. Bigo : (1) 42 80 02 35.

Vends photocopieuse 3M type 207, TBE avec meuble sacrifié, 2900F. Tél : 40 50 01 71.

Vends disque dur SH205 : 2700F, émulateur PC Supercharger 1 Mo de mémoire : 2300F, lecteur 5"1/4 40/80 pistes externe : 1000F, Turbo C pour PC français ou international : 500F, Paradox 3 PC complet : 3500F. Tél : 30 76 07 72, le soir.

Echange Amiga 1000 contre Atari 1040 STE (en bon état) ou vends à 3900F, cause musicien en pénurie de logiciels ! Tél : 69 38 52 40, demander Laurent (dans le 91).

Vends Atari 1040 ST, moniteur, imprimante, log TDT, souris : 5000F. Tél : 48 26 04 80 après 18h.

Vends Atari 520 STE (garanti jusqu'en Jan. 91), souris, jeux, utilitaires, le tout en très bon état : 2700F à débattre. Peut se déplacer sur Paris et banlieue proche. Tél : 43 06 41 88 à partir de 17h30, demander Martin.

Vends Atari Mega ST 1 monochrome, Citizen 120 D, GfA 3, 6 livres sur GfA, le Rédacteur, Timeworks, flexdisc, jeux, joystick, avec ST Mag numéros 9 à 35. Le tout 9000F. Tél : Yann au 48 57 73 27.

Urgent ! Vends Atari 1040, lecteur double face, livres, utilitaires, jeux, GfA Basic et bien d'autres... Le tout sacrifié pour 2000F. Alamaru, Tél : 47 83 66 65 ou 43 99 99 66.

Vends Amiga 2000 (89), écran couleur 1084, carte PC XT, disque dur 20 Mo, peu servi : 15000F. Tél : 47 51 00 19.

Vends Atari Mega ST 1, moniteur monochrome SM 124, imprimante Epson LX 800, programmes, le tout : 7500F. Contacter M. Narcy : (dom) 45 35 88 79, (job) 45 58 34 24.

Vends Atari 520 ST + joystick, état neuf, 1900F. Tél : (1) 69 06 24 52.

Vends 1040 STF (août 88), étendu à 3 Mo, blitter, horloge, ressorts clavier, état neuf : 4500F. Vends disque dur SH205, parfait état : 2700F. Sébastien, au 42 45 10 12.

A vendre logiciel Superbase Professional 3, achat 5/90, prix : 2490F, vendu 2000F, cause erreur achat. Tél : 60 86 63 14 après 19h.

Vends Mega ST 2, SM 124, SH 205 (HD 20Mo), Mark Williams C, Minix (Unix en source), Joystick. parfait état. L'ensemble 10000F. Tél : 42 72 97 79, après 19h.

Le roi des logiciels de traitement de texte : Signum 2! Vendu pas cher cause double emploi : 700F. Olivier, au 40 44 65 26.

Stop affaire ! Vends softs : Dungeon Master (150F), Bloodwich 1 et 2 (150F chaque), Oblitérator (100F), Infestation (200F), Hard Driving (200F) et utilitaires : Dali 3 (450F), STOS (600F), Spack (250F), Turbo ST (250F), Script (300F), The Killer (200F), Oss Pascal (300F). Tél : 47 32 41 19 ou 47 52 11 82.

Vends originaux état neuf pour ST : Adimens 400F, LDW Power 700F, Script 400F, ZZ-Rough 1.1 200F, Multiface ST 400F. Tél : 64 66 23 62.
Vends Atari 1040 ST + moniteur couleur + imprimante Mannessmann 81 + joystick + logiciels. Très peu servi encore garanti, prix : 5500F. Tél : 48 95 82 96.

Vends 520 ST étendu à 1 Mo : 1700F, lecteur externe SF 314 DF : 1000F, moniteur mono SM 124 et inverseur mono/couleur : 1000F, imprimante Epson LX800 avec tracteur listing 1600F. Tél : 34 69 28 40.

Vends Amiga 2000 (moniteur A1084, UC, clavier, DD 30 Mo, XT, lecteur 5"1/4, tte la documentation d'origine et les logiciels + logiciels de jeux), 1 an et demi, très bon état, 17000F. Tél : 43 04 35 00, après 20h, Ahmadou.

Vends Mega ST 4, Megafile 30, lecteur externe, Nec P2200, Aladin 3.0, Exchanger, Handy Scanner, ensemble de logiciels originaux (PAO : Calamus et PPM, CAO, puis des jeux...), le tout : 25000F. Tél : 40 37 36 76.

VENTES - PROVINCE

Cause achat Mac, vends disque dur Megafile 30 (neuf, janvier 90) : 4000F. Imprimante Seikosha SL80AI 24 aiguilles : 2800F, logiciels Le Rédacteur 3 (neuf) : 800F, Calligrapher Pro (neuf) : 1000F, Publishing Partner Master (jamais servi) : 2000F, G+PLUS, Induction, Calcomat 2. Tél : 99 31 17 25.

Vends imprimante Mannesmann Tally MT81 TBE, encore sous garantie jusqu'au 15/1/91 : 1000F, nombreux livres sur le ST de chez Micro-Application, free-boot externe : 50F, joystick infrarouge : 100F. Tél : 40 20 34 24 ou 51 22 05 05.

Vends originaux Marble Madness : 100F, Silent Servie : 150F, Explora : 150F, BEA : 150F, Mindfighter : 200F, Twist : 200F, Image Scan : 1000F, ZZ-Lazy Paint : 600F. Après 19h, Tél : 99 51 64 37.

Vends Interface DMA/SCSI de DCI, avec soft, garanti, neuf, jamais servi : 950F. Vends original du GfA Basic 3.0 + mise à jour 3.5 : 550F. Tél : Jean-Pierre au 50 49 09 04 après 18h30.

Vends RAMs 41256 à 100ns : 30F pièce (moins si grande quantité). Tél : 35 87 13 78 (après le 27 juillet).

Vends 50 fontes GDOS soignées, 94 caractères, hauteur 30 à 60 points, classiques et fantaisie. Prix : 5F la fonte, plus disquette. Hardcopy sur demande. Tél le matin : 87 86 68 09.

A vendre Mega ST 2, moniteur mono, imprimante Star LC-10, 40 jeux, joystick, langages (C, Basic), PAO-DAO : 10000F. Tél : 67 76 23 87 après 18h.

Grande Braderie : Atari Mega 4E : 12960F, STE 4 Mo : 6200F, STE 2 Mo : 4800F, moniteur multisync couleur 800x600 avec interface Atari : 4200F, disque dur SCSI 40 Mo : 5600F, extensions mémoire SIMM/SIP à 1 Mo : 400F, à 2 Mo : 2500F (tous sous garantie), laser 6 pages/mn : 6800 F (occase). Tél : 56 92 03 02.

A vendre 1040 ST + SM 124 TBE 3900F, imprimante STAR LC24/10 (Fév 90) très peu servie, sacrifiée : 2200F, expandeur Kawai PHM 200 sons, poly 30 voies, multi 4 voies + drum kit : 1850F. Logiciels (originaux) Mégapage (avr 90) : 600F, émulateur PC Ditto avec carte enregistrement 400F, GfA Raytrace 270F. Expéditions possibles. Tél : 58 41 55 19 (sud-ouest).

Scanner A4 à plat (imprimante, photocopieuse), peu utilisé avec soft et programme "Image Partner" original : 5500F (port compris). Ecrire à F.Grenier, 10 rue Curtat, 1005-Lausanne, Suisse.

Vends originaux pour Atari ST : Ninja Warriors, Robocop, Chase HQ, Ivanhoé, Rainbow Island, Chamber of Shaolin, Strider, Dragon Ninja, Hard Drivin, Ninja Spirit, Speedball, F29, Xenon 2, Tennis Cup, de 79 à 94 francs, et BAT, 250F. Tél : 99 52 06 78 ou 40 34 35 79.

Vends Aladin complet, version 3.0 dans sa boîte d'origine 1500F. Tél : 99 48 02 52.

Vends banque de rythmes format Pro 24 en deux disquettes : Breaks + Rythmes : 250F contenant plus de 1000 patterns pour D110 Roland. Masson Marc, 3 allée de Picardie, 54390 Frouard.

ACHATS - PROVINCE

Achète Atari 1040 STE ou Mega ST 1 + écran couleur + souris + logiciels originaux. Etat neuf pour 4800F. Contacter d'urgence Mathieu au 62 96 03 26 après 20h.

EMPLOIS

Application Systems recherche des technico-commerciaux. Téléphoner au (1) 40 92 80 81 pour rendez-vous. Poste basé à Montrouge.

Cherche personne sur Paris pour diriger un journal. Pour tout renseignement, tél : (1) 46 87 23 15, demandez Guerchon, laissez un message si absent.

DIVERS

Recherche plans complets 520 STF. Faire offre au (1) 30 57 13 10 après 19h.

Cherche local sur Paris. Pour toutes propositions, écrire à M. Guerchon Patrick, BP 78, 94320 Thiais.

Recherche programmeurs en assembleur, graphistes, musiciens sur ST pour faire des démos. Berogin Sébastien, 18 rue de la Liberté, 54300 Chanteheux. Tél : 83 73 70 77.

Pour passer une annonce, prenez le bulletin du mois dernier, joignez-y 50 F à l'ordre de Pressimage (ou 25 pour les abonnés). Envoyez le tout à ST Magazine-PA, 19 rue Hégésippe Moreau, 75018 Paris, avant le 20 août pour la parution du 10 septembre. Ouf !

LA BOUTIQUE DE PRESSIMAGE

La Boutique sera fermée du 15 juillet au 15 août, il est donc inutile de commander pendant cette période car les commandes ne seront servies qu'à partir du 15 août. Pour nous faire pardonner ce long, mais cependant indispensable, mois d'absence, vous aurez la joie de découvrir dans le numéro de septembre, notre nouveau catalogue bourré de surprises et de nouveautés !

LES NOUVEAUTÉS DU MOIS

"LE JOUEUR DE CARTES"

Tous modèles de ST. Couleur uniquement.
75 francs.
Le Joueur de Cartes vous propose de jouer à trois variantes de réussite sans avoir à mélanger les cartes, à les distribuer, à les ramasser et recommencer le tout. Le programme se charge de tout, il vous suffit de cliquer !
La première variante s'appelle REUSSITE.
Les cartes sont réparties sur sept colonnes. Le but de la partie est de les ranger par couleur dans l'ordre croissant en commençant par l'As.
La seconde porte le nom de PATIENTE.
Vous devez regrouper les cartes en un minimum de paquets. Les cartes sont distribuées une à une de gauche à droite à la première ligne, puis en sens inverse à la ligne suivante, etc. Vous pouvez opérer un regroupement lorsque deux cartes de même couleur ou de même valeur ne sont séparées que par une seule carte. Cliquez sur la carte du milieu : elle recouvre celle qui la précède et toutes celles qui suivent sont décalées d'un cran.
La troisième réussite est l'une des plus connues, c'est le SOLITAIRE.
Le plateau de jeu est composé de 4 colonnes, sur lesquelles les cartes sont distribuées 4 par 4. Lorsque 2 des cartes visibles sont de la même couleur, la carte la plus faible est éliminée. La carte de plus forte valeur pour cette réussite est l'As. Pour gagner vous devez terminer la réussite avec seulement un As dans chaque colonne.
Vous possédez 500 picaillons (la monnaie locale) au début du jeu. Chaque fois que vous commencez une réussite, vous devez miser une somme variant entre 50 et 300 picaillons. Les gains diffèrent selon les variantes.
Une excellente compilation avec des écrans de jeux superbement réalisés.

"U-220 MIDI PROGRAMMER"

Pour tous modèles de ST sauf les 520.
(minimum 1 mégaoctet de mémoire)
Monochrone uniquement.
195 francs.
Le U220 Midi Programmer permet l'édition et la sauvegarde de l'ensemble des paramètres éditables de l'expandeur Roland U-220. Ce logiciel est conçu pour être utilisé conjointement avec un clavier maître. La transmission des données dans le sens U-220 -> ST n'est nécessaire que lors du lancement du programme (phase d'initialisation) des informations provenant d'un clavier externe. Les principales caractéristiques de cet éditeur

BON DE COMMANDE

PRODUITS	QTÉ	PRIX
FRAIS DE PORT (15 francs, sauf indications contraires)		
NET A PAYER		

LOGICIELS ATARI ❑
LOGICIELS AMIGA ❑

BON DE COMMANDE A RETOURNER A:
LIBRAIRIE PRESSIMAGE
210 rue du Faubourg St MARTIN
75010 PARIS

J'ai connaissance du fait que certains produits ont une notice ou un écran en langue anglaise et je désire les acquérir.
Veuillez expédier ma commande à l'adresse suivante :

NOM :...

PRENOM :...

ADRESSE :...

....

CODE POSTAL :....................................

VILLE :...

DATE : ....../....../1990

SIGNATURE : (celle des parents pour mineurs)

Ci-joint mon règlement à l'ordre de PRESSIMAGE

❑ Chèque
❑ C.C.P

Etranger, nous consulter.

BON DE COMMANDE

PRODUITS	QTÉ	PRIX
FRAIS DE PORT (15 francs, sauf indications contraires)		
NET A PAYER		

LOGICIELS ATARI ❑
LOGICIELS AMIGA ❑

BON DE COMMANDE A RETOURNER A:
LIBRAIRIE PRESSIMAGE
210 rue du Faubourg St MARTIN
75010 PARIS

J'ai connaissance du fait que certains produits ont une notice ou un écran en langue anglaise et je désire les acquérir. Veuillez expédier ma commande à l'adresse suivante:

NOM:.....

PRENOM:.....

ADRESSE:.....

.....

CODE POSTAL:.....

VILLE:.....

DATE: ...../...../1990

SIGNATURE: (celle des parents pour mineurs)

Ci-joint mon règlement à l'ordre de PRESSIMAGE

❑ Chèque
❑ C.C.P

Etranger, nous consulter.

sont : l'édition des timbres, gestion des banques de timbres, édition des patches, gestion des banques de patches, édition des configurations rythmiques (rhythm set), édition des tableaux de correspondance (maps), édition de la configuration générale (Setup), fonctions de dump (timbre, patch, rhythm set, map, all datas), transmission immédiate des paramètres... La convivialité du U220 Midi Programmer vous permettra d'utiliser facilement les immenses possibilités de votre expandeur et d'éviter les fastidieuses séances de réglages directement sur le U220.

"ASSEMBLEUR N° 2"
Pour AMIGA. **75 francs.**
Cette disquette regroupe certains des listings parus dans le cahier Amiga de ST Magazine à partir du numéro 38, certains listings n'ayant pas encore été intégralement publiés. Elle contient les codes sources complets en assembleur et en Gfa 3 des programmes : TURMITES (ST Mag 40 et 41), PALETTES (ST Mag 38) ainsi que DESSASEMBLEUR (avec version compilée) et DEBUGGER.

"DISCPAC"
Pour tous modèles de ST. Monochrome et couleur.
20 francs.
DISCPAC est ce qu'on appelle souvent un "éditeur de secteurs". Il permet de visualiser et de modifier (éditer, donc) le contenu d'une disquette, qui est organisée en pistes, elles-mêmes divisées en secteurs. DISCPAC est composé de 2 parties principales. La partie supérieure de l'écran permet de visualiser et d'éditer le "secteur" en cours, la partie inférieure permet de choisir le secteur que l'on désire éditer, d'effectuer des recherches, de changer de mode, de définir des paramètres, ou encore de quitter le programme. DISCPAC peut travailler dans trois modes différents: PHYSIQUE, LOGIQUE, et FICHIER. Après avoir choisi un mode d'édition, un lecteur et un secteur à éditer, le secteur est automatiquement affiché, repérant en permanence la position du curseur (rectangle clignotant), et la représentation en ASCII, hexadécimal, décimal et binaire de l'octet correspondant. Il arrive souvent que l'on sache quoi modifier (par exemple le nom des meilleurs joueurs dans une table de Hi-Scores), mais que l'on ne sache pas où cela se trouve précisément. On utilise alors la fonction recherche, qui permet de spécifier une chaîne de caractères que DISCPAC cherchera sur tout le disque. Si vous avez des questions, remarques, suggestions, critiques, n'hésitez pas à déposer un mot dans la BAL STJC sur les serveurs de Pressimage, accessibles par le 3615 STMAG ou le 3615 GEN4.

"TRACK-BALL" du nouveau..."
Vous pouvez maintenant vous procurer notre fantastique et géniale super souris en deux modèles différents. ***TRACK-BALL SOURIS*** : c'est tout simplement une méga-souris à deux boutons au prix de ***290 francs.***
Vous pouvez également vous procurer le ***TRACK-BALL JOYSTICK***, qui, lui, émule en plus de la souris les fonctions d'un joystick. Il est disponible pour ***345 francs.***

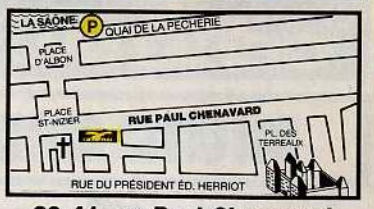
PARIS

10, boulevard de Strasbourg
75010 PARIS ☎ 42.06.50.50
ouvert tous les jours sauf dimanche
SAV : 54, rue René-Boulanger
75010 PARIS ☎ 42.06.77.78
ouvert du mardi au samedi
métro Strasbourg/St-Denis - Parking à proximité
Magasins ouverts de 9 h 45 à 13 h et de 14 h à 19 h

GENERAL®

LE GRAND SPECIALISTE INFORMATIQUE

LYON



39-41, rue Paul-Chenavard
69001 LYON ☎ 72.00.96.96
Pour tout achat, GVL vous offre 1 heure de stationnement sur présentation du ticket de parking du Quai de la Pecherie
ouvert du mardi au samedi
de 9 h 45 à 13 h et de 14 h à 19 h

ACHETER UN ATARI CHEZ GENERAL, C'EST ACHETER CHEZ LE PLUS IMPORTANT SPÉCIALISTE INFORMATIQUE INDEPENDANT EN FRANCE

DISQUETTE 3"1/2 DF-DD 4,90F TTC pièce

LECTEUR 3"1/2 POUR ATARI 790F TTC

LECTEUR 5"1/4 POUR ATARI 990F TTC

GENERAL vous offre votre ATARI 520 STE

CONFIGURATION MIDI COMPLETE
9990F TTC

En cadeau : VOTRE ATARI 520 STE + ST REPLAY

CLAVIER PC2000 DYNAMIC 6 OCTAVES + SEQUENCEUR 10 PISTES TENTRAX + EXPENDEUR ROLAND CM32L, 128 SONS, REVERB + MINI MONITEUR AUDIO AMPLIFIER 2xMA12C

ATARI PC AT ABC 286 EGA

9990F TTC

- 640 Ko extensible à 4 Mo
- Disque dur 30 Mo
- Microprocesseur 80286
- Lecteur 3"1/2 1,44 Mo
- Clavier 102 touches
- 3 slots d'extension
- Interfaces série et parallèle

pour commander, se renseigner, jouer et gagner, c'est simple : tapez

3615 code GVF

CONSOLE ATARI LYNX

+ 4 JEUX 1490F TTC

OPERATION VACANCES

achetez votre matériel maintenant et PAYEZ en OCTOBRE*

* Achat minimum 2000 F, après acceptation du dossier

PARIS 10e
Tél. 42.06.50.50
LYON 1er
Tél. 72.00.96.96

LE GRAND SPECIALISTE ATARI

MICRO ORDINATEUR PERSONNEL ATARI 520 STE

CARACTERISTIQUES TECHNIQUES

Microprocesseur : 68000 (16/32 bits)
Mémoire vive : 512 Ko extensible à 4 Mo
Mémoire de masse : 1 lecteur 3 pouces 1/2 720 Ko
Son et Musique : Midi et son stéréo 8 bits
Résolution maxi : 640 x 400 pixels
Palette couleur : 16 parmi 4096 couleurs
Système : GEM (graphiques et souris)

Depuis son introduction, le 520 ST s'est imposé comme l'incarnation de la technologie de pointe à la portée de tous : processeur 16 / 32 bits, puissance de traitement, rapidité d'exécution, capacité mémoire, ouverture vers les périphériques. Le tout maîtrisé grâce à la convivialité et la simplicité de l'environnement graphique GEM et de la souris.

Son avance technologique et la richesse de sa bibliothèque de logiciels ont fait du 520 ST un micro fascinant dans tous les domaines : création graphique, musique, éducation, programmation et jeux.

Pour offrir encore plus de performances, ATARI lance aujourd'hui le 520 STE. La couleur, le son, les interfaces, la mémoire, la vidéo, le système d'exploitation, autant d'innovations pour autant de domaines d'applications. Le STE est résolument la machine de tous les loisirs. Jamais le grand public n'a eu autant d'atouts technologiques entre ses mains, pour son plus grand plaisir. Lorsqu'on sait que malgré ses fantastiques nouvelles possibilités, le prix du STE reste inchangé par rapport à celui du ST, il n'y a plus d'alternative dans le choix d'un micro-ordinateur de loisir.

ATARI 520 STE **3490F TTC**	A crédit CETELEM : 90F au comptant + 18 mensualités de 227,90F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 820,40F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 520 STE + Monit. mono SM124 **4490F TTC**	A crédit CETELEM : 90F au comptant + 24 mensualités de 231,40F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 1273,60F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 520 STE + Monit. Couleur SC 1425 **5490F TTC**	A crédit CETELEM : 90F au comptant + 36 mensualités de 206,10F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2139,60F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 520 STE + Moniteur mono SM 124 + Imp. Mannes. Tally MT81 **5790F TTC**	A crédit CETELEM : 90F au comptant + 36 mensualités de 217,50F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2250F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 520 STE + Monit. couleur SC 1425 + Imp. Mannes. Tally MT 81 **6890F TTC**	A crédit CETELEM : 90F au comptant + 48 mensualités de 211,50F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 3472F dont 120F de perception forfaitaire - TEG 17,92

MICRO ORDINATEUR PROFESSIONNEL ATARI 1040 STE

CARACTERISTIQUES TECHNIQUES

Microprocesseur : 68000 (16/32 bits)
Mémoire vive : 1 Mo extensible à 4 Mo
Mémoire de masse : 1 lecteur 3 pouces 1/2 720 Ko
Son et Musique : Midi et son stéréo 8 bits
Résolution maxi : 640 x 400 pixels
Palette couleur : 16 parmi 4096 couleurs
Système : GEM (graphiques et souris)

L'ATARI 1040 STE intègre les toutes dernières innovations de la technologie de pointe, aussi bien au niveau des composants que des méthodes de production. Ainsi l'ATARI 1040 STE profite des plus récentes découvertes de la micro-électronique : architecture résolument innovatrice basée sur le microprocesseur 16/32 bits MC 68000 et des coprocesseurs spécialisés à très haute intégration.
Résultat : des performances en hausse et des coûts de production en baisse ; ce qui permet de vous offrir un micro-ordinateur professionnel à un prix ultra-compétitif. Le choix d'un ordinateur, pour son usage personnel dans le cadre professionnel, ou pour la gestion de l'entreprise, est aujourd'hui plus facile avec l'ATARI 1040 STE.
Avec toute sa puissance, offerte à un prix très attractif, l'ATARI 1040 STE se place en tête du rapport performances/coût. Voilà pourquoi il a été élu ordinateur de l'année aux U.S.A., qu'il est best-seller en Allemagne et qu'il fait déjà la une de la presse professionnelle en France. Plus de 2000 développeurs dans le monde - plus de 150 en France - créent des logiciels pour l'ATARI ST. Le langage de développement d'applications Memsoft est en standard sur l'ATARI ST. De ce fait, plusieurs dizaines de solutions de gestion et d'applications verticales pour PME/PMI sont rendues opérationnelles sur ATARI.

ATARI 1040 STE **4490F TTC**	A crédit CETELEM : 90F au comptant + 24 mensualités de 231,40F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 1273,40F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 1040 STE + Monit. mono SM124 **5490F TTC**	A crédit CETELEM : 90F au comptant + 36 mensualités de 206,10F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2139,60F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 1040 STE + Monit. Couleur SC 1224 **6490F TTC**	A crédit CETELEM : 90F au comptant + 36 mensualités de 244,30F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2514,80F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 1040 STE + Moniteur mono SM 124 + Imp. Mannes. Tally MT81 **6490F TTC**	A crédit CETELEM : 90F au comptant + 36 mensualités de 244,30F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2514,80F dont 120F de perception forfaitaire - TEG : 17,92 %
ATARI 1040 STE + Monit. couleur SC 1224 + Imp. Mannes. Tally MT 81 **7490F TTC**	A crédit CETELEM : 90F au comptant + 48 mensualités de 230,10F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 3764,80F dont 120F de perception forfaitaire - TEG 17,92

PARIS 10e
Tél. 42.06.50.50
LYON 1er
Tél. 72.00.96.96

LE GRAND SPECIALISTE ATARI

ORDINATEUR PROFESSIONNEL ATARI MEGA ST

CARACTERISTIQUES TECHNIQUES

Microprocesseur 16/32 bits, MC 68000 à 8 MHz. Coprocesseur Blitter Atari. Horloge interne sauvegardée par piles. Système d'exploitation GEM de Digital Research. Graphisme haute définition 640 × 400. Moniteur monochrome haute résolution. Clavier détaché ergonomique AZERTY accentué. Bloc numérique 10 touches séparé. Souris. Lecteur de disquettes intégré, 3 pouces 1/2, 720 Ko. Nombreuses interfaces en standard : RS 232, Centronics, DMA 10 mégabits/seconde pour disque dur, émulateur VT 52.

LES CARACTERISTIQUES : La gamme MEGA ST représente le summum du concept ST, avec beaucoup de RAMS, un graphisme ultra-performant et un DOS arrivé à maturité. ATARI fonde de grands espoirs sur cette gamme pour l'édition assistée par ordinateur.

ASPECT TECHNIQUE : Le MEGA ST est tout petit, bien que le clavier soit plus grand que celui d'un 1040. Nous sommes en effet si habitué à la dimension d'un PC. La hauteur du CPU n'est que de 2 pouces 1/2. Le clavier, avec ses touches munies de LED, de couleur gris clair, ainsi que le CPU confèrent à cette configuration un esprit de famille résolument ATARI. Il faut dire qu'avec le moniteur ATARI plein écran monté sur le CPU, l'ensemble a un air très professionnel. Comme la gamme ST, la gamme MEGA ST est chargée de prises en tous genres. Qu'on en juge : Port série RS232 25 connecteurs, port imprimante parallèle Centronics, entrée midi, sortie midi, sortie vidéo, port floppy disk extérieur, port DMA (Direct Memory Access, accès direct mémoire). Vous trouverez aussi au dos de l'appareil un bouton de reset, l'entrée secteur, le ventilateur et un vide avec un panneau l'obstruant sur lequel il est inscrit "extension". Sur le côté droit, il y a une ouverture qui contient le port cartouche (jusqu'à 128 Ko de ROM) et la prise clavier. Les autres caractéristiques externes intéressantes sont les deux ports joystick/souris à l'arrière du clavier et la fente du floppy disk sur la façade avant.

UN OUTIL PROFESSIONNEL

Outil professionnel par excellence, l'ATARI MEGA ST 1 est particulièrement bien adapté, avec sa fantastique puissance, à tous les travaux demandant un graphisme soigné, en couleur et en monochrome : bureautique, création graphique, animation d'images, CAO/DAO, etc... Associé au disque dur ATARI SH 205 – MEGA FILE, le MEGA ST 1 devient la machine de référence dans le domaine des solutions de gestion ; doté de l'imprimante laser ATARI SLM 804, le MEGA ST 1 constitue l'ensemble PAO le plus attractif du marché.

PERFORMANCES GRAPHIQUES DE TRÈS HAUT NIVEAU

L'ATARI MEGA ST 1 est doté de composants très élaborés qui lui permettent de parvenir à des performances graphiques de très haut niveau.

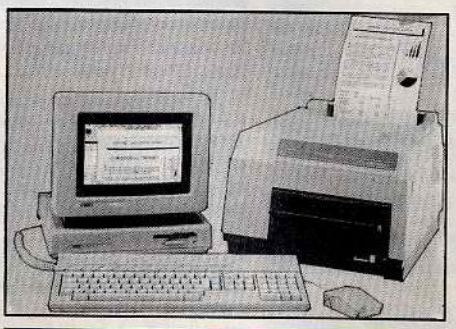
BUS COMPLET 68000

OUVERTURE TOTALE

Un bus complet 68000 équipe l'ATARI MEGA ST 1 et l'ouvre totalement vers le monde extérieur. En conséquence, une multitude d'applications devient envisageable, notamment dans les domaines scientifique et technique : systèmes d'acquisition de données, systèmes d'entrée-sortie numériques, systèmes d'entrée-sortie industriels, périphériques (écran très haute résolution, liaison IEEE...), cartes mémoires RAM et ROM additionnelles, cartes munies d'autres microprocesseurs et bien évidemment les bus standards (type VME par exemple).

10 INTERFACES EN STANDARD

Les nombreuses interfaces qui équipent en standard le MEGA ST 1 lui procurent une ouverture vers un environnement diversifié en permettant de connecter la plupart des périphériques : disques durs, imprimantes, modems, synthétiseurs, etc.



MEGA ST4 + MONIT. SM 124 + IMP. LASER SLM 804 + PUBLISHING PARTNER + DISK DUR MEGAFILE 30
32615F TTC
A crédit CETELEM : 615F au comptant + 48 mensualités de 983,90F - 1re échéance 90 jours après achat - Coût total du crédit avec assurance : 15467,20F dont 240F de perception forfaitaire - TEG : 17,35 %

ATARI MEGA PAGE (Mega ST 1 + Monit. SM 124 + Timeworks + Rédacteur)
6490F TTC
A crédit CETELEM : 90F au comptant + 36 mensualités de 244,30F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 2514,80F dont 120F de perception forfaitaire - TEG : 17,92 %

Atari MEGAPAGE Disk Dur (Mega ST1 + Monit. SM 124 + Timeworks + Rédacteur + Disk Dur 30 Mo)
9850F TTC
A crédit CETELEM : 50F au comptant + 48 mensualités de 304,70F 1re échéance 90 jours après achat Coût total du crédit avec assurance : 4945,60F dont 120F de perception forfaitaire - TEG : 17,92 %

IMP. LASER SLM 804
13579F TTC
A crédit CETELEM : 579F au comptant + 48 mensualités de 404,20F - 1re échéance 90 jours après achat - Coût total du crédit avec assurance : 6641,60F dont 240F de perception forfaitaire - TEG : 17,92 %

IMPRIMANTE LASER SLM 804

Vitesse d'impression : 8 pages par minutes. Résolution : 300 × 300 points au pouce. Format d'impression : A4, à la française ou à l'italienne. Chargement du papier : automatique ou manuel. Capacité du magasin de papier : 150 feuilles. Interface : port DMA du MEGA ST 1. Impression mode texte : émulation Diablo 630. Impression mode graphique : GDOS de Digital Research, logiciel de composition de page en cours de développement.

ATARI MEGA ST 4 + Monit. mono Atari SM 124
14765F TTC
A crédit CETELEM : 265F au comptant + 48 mensualités de 450,80F - 1re échéance 90 jours après achat - Coût total du crédit avec assurance : 7378,40F dont 240F de perception forfaitaire - TEG : 17,92 %

ATARI MEGA ST 4 + Monit. coul. Atari SC 1425
15765F TTC
A crédit CETELEM : 265F au comptant + 48 mensualités de 478,90F - 1re échéance 90 jours après achat - Coût total du crédit avec assurance : 7727,20F dont 240F de perception forfaitaire - TEG : 17,60 %

ATARI MEGA ST 4 + Monit mono Atari SM 124 + Imp. laser Atari SLM 804
26388F TTC
A crédit CETELEM : 388F au comptant + 48 mensualités de 799,40F - 1re échéance 90 jours après achat - Coût total du crédit avec assurance : 12611,20F dont 240F de perception forfaitaire - TEG : 17,35 %

PORTABLE ATARI STACY

Pour la première fois, on dispose, avec le STACY, dans une unité centrale une grande capacité mémoire en standard (jusqu'à 4 Mo de RAM) pleinement utilisable dans un environnement graphique avec écran haute résolution.

CARACTERISTIQUES TECHNIQUES

MICROPROCESSEUR : Motorola 68HC000 (version CMOS du 68000) cadencé à 8 MHz.

SYSTEME D'EXPLOITATION : TOS et environnement graphique GEM en ROM. Format de disquette 3.5" compatible MS-DOS autorisant des transferts de fichiers sur compatibles PC.

MEMOIRE : ROM 256 Ko. RAM 4 Mo (LST 4144).

MEMOIRE DE MASSE : lecteur de disquette interne 3.5" de 720 Ko.

RESOLUTIONS GRAPHIQUES : haute résolution monochrome (écran LCD ou externe) ; moyenne résolution couleur (écran externe ou TV) ; basse résolution couleur (écran externe ou TV).

INTERFACES : prises MIDI IN et MIDI OUT/THRU ; prise moniteur (RVB analogique, haute résolution monochrome ou audio) ; port parallèle (imprimante) ; port série RS 232C ; port unité de disquette externe ; port DMA ; port cartouche (128 Ko) ; port souris ; port manette.

ALIMENTATION : adaptateur secteur externe 18 V 2 A. Piles 1.5 V de type LR 14 ou accus rechargeables.

DIMENSIONS : largeur 38,5 cm, profondeur 33 cm, épaisseur 8,7 cm. POIDS : 5,5 kg.

23490F TTC

PARIS 10e
☎ 42.06.50.50
LYON 1er
☎ 72.00.96.96

LE GRAND SPECIALISTE MICRO

LES IMPRIMANTES

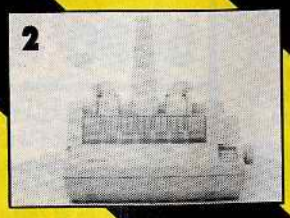
Toutes les imprimantes sont garanties 2 ans*
Cadeau PACK PRO IMPRIMANTES GENERAL
* Sauf tête d'impression

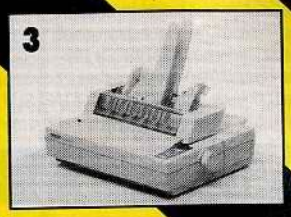
PROMOTION SUR NOS RUBANS

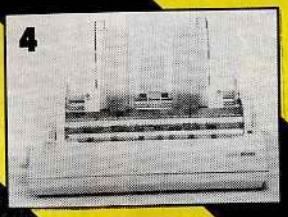
à partir de 50F TTC pièce

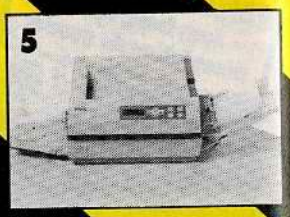
Nous tenons à votre disposition un stock énorme de rubans à des prix très compétitifs. Consultez-nous !

1

2


3


4


5


18

17

16

15

AMSTRAD

FIRST PUBLISHER est un logiciel de Publication Assistée par Ordinateur. Il fonctionne sur tout PC, PC-XT, PC-AT, PS2 et compatibles. Ce logiciel réalise la mise en page automatique sur 1 à 4 colonnes, vous disposez d'une bibliothèque de 150 dessins et d'outils de dessins (crayon, gomme...). FIRST PUBLISHER possède de nombreuses fonctions de traitement de texte ainsi que 60 polices de caractères différentes avec leurs attributs. **Valeur : 1482,50FTTC.**

DMP 3160 (160 cps - 9 aiguilles - 80 col)	(1)	1790 F
DMP 3250 (160 cps - 80 col - PAR/SER) + **First Publisher**		1990 F
LQ 3500 (160 cps - 24 aiguilles - 80 col) + **First Publisher**		2690 F
LQ 5000 (288 cps - 24 aiguilles - 132 col) +**First Publisher**		4990 F

EPSON

LX 800 (180 cps - 9 aiguilles - 80 col)	(2)	2150 F
LX 850 (200 cps - 9 aiguilles - 80 col)		2690 F
LQ 500 (180 cps - 24 aiguilles - 80 col)	(3)	3695 F
LQ 550 (226 cps - 24 aiguilles - 80 col)		3990 F
FX 850 (264 cps - 9 aiguilles - 80 col)		4990 F
IX 800 (375 cps - 9 aiguilles - 80 col)		5750 F
FX 1050 (264 cps - 9 aiguilles - 132 col)		5990 F
LQ 850 (330 cps - 24 aiguilles - 80 col)		6590 F
LQ 1050 (330 cps - 24 aiguilles - 132 col)	(4)	7690 F
LQ 860 (300 cps - 24 aiguilles - 80 col - 256 couleurs)		8290 F
LQ 1060 (300 cps - 24 aiguilles - 132 col - 256 couleurs)		9490 F
SQ 850 (742 cps - 24 aiguilles - 80 col - jet d'encre)		6990 F
LQ 2550 (500 cps - 24 aiguilles - 132 col) avec kit couleur		11600 F
SQ 2550 (743 cps - 24 aiguilles - 132 col - jet d'encre)		10990 F
DFX 5000 (400 LPM - 9 aiguilles - 132 col)		19390 F
GQ 5000 Laser	(5)	14990 F
TSQ 4800		14290 F
TLQ 4800		18290 F
QX 800 (Jet d'encre coul. + videur d'écran CGA coul.)		13990 F

NEC

P2 PLUS(96 cps - 24 aiguilles - 80 col)	(6)	3690 F
P6 + (264 cps - 24 aiguilles - 80 col)	(7)	5990 F
P7 + (264 cps - 24 aiguilles - 132 col)		7990 F

* sauf tête d'impression

MANNESMANN TALLY

MT81 (155 cps - 9 aiguilles)	(8)	1590 F
MT 222 (264 cps - 24 aiguilles)		5890 F
MT 905 laser	(9)	13990 F
MT 81 MINITEL (idéale pour vos copies d'écran)		2490 F

CITIZEN

120 D+(120 cps - 9 aiguilles - 80 col)	(10)	1290 F
HQP 40 (200 cps - 24 aiguilles - 80 col)	(11)	PROMOTION : 2990 F
MSP 15 E (160 cps - 9 aiguilles - 132 col)		PROMOTION : 2690 F
SWIFT 24 (160 cps - 24 aiguilles - 80 col.)	(12)	3490 F
Kit Couleur SWIFT 24		590 F
MSP 40 (240 cps - 9 aiguilles - 80 col)		4490 F
HQ P 45 (200 cps - 24 aiguilles - 132 col)		4990 F
MSP 50 (300 cps - 9 aiguilles - 80 col)		5490 F
MSP 55 (300 cps - 9 aiguilles - 132 col)		6490 F

STAR

LC 10 (144 cps - 9 aiguilles - 80 col)	(13)	1890 F
LC 10 couleur (144 cps - 9 aiguilles - 80 col)	(13)	2490 F
LC 2410 (170 cps - 24 aiguilles - 80 col)	(14)	2990 F
LC 15 (9 aiguilles)		3990 F
LC 2415	(15)	4990 F
FR 10 (300 cps - 9 aiguilles - 80 col)		4990 F
FR 15 250 cps - 9 aiguilles - 136 col)		6190 F
XB 24-10 (240 cps - 24 aiguilles - 80 col)		6690 F
XB 24-15 (200 cps - 24 aiguilles - 136 col)	(16)	7900 F
Laser Printer LP 8		19990 F

PANASONIC

KXP 1180 (182 cps - 9 aiguilles - 80 col)	(17)	2490 F
KXP 1124 (192 cps - 24 aiguilles - 80 col)		3990 F

INTRADE

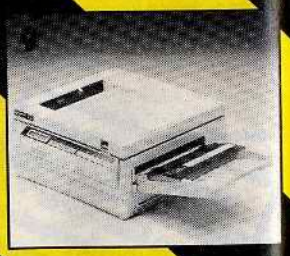
E 2410 (216 cps - 24 aiguilles - 80 col)	(18)	2990 F
E 2415 (216 cps - 24 aiguilles - 132 col)		4990 F

Nos prix sont annoncés toutes taxes comprises

6

7

8

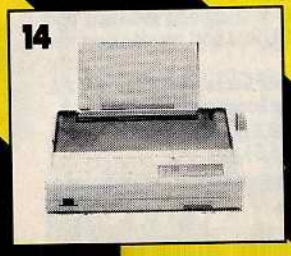
9


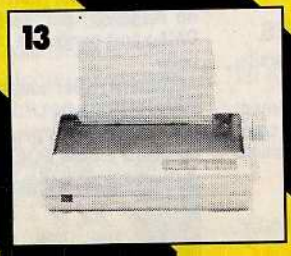
-30 % SUR VOTRE IMPRIMANTE

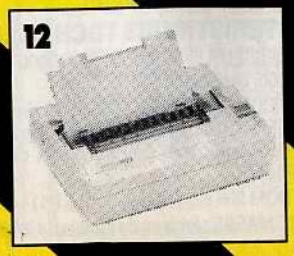
pour l'achat groupé de certaines imprimantes
avec un ordinateur d'une valeur supérieure à 5000 F (sauf promotions)

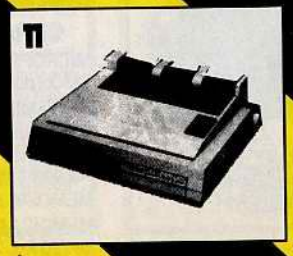
EXEMPLE POUR UN ACHAT GROUPÉ :

MANNESMANN TALLY MT 81	1110 F TTC
STAR LC 10	1320 F TTC
EPSON LX 800	1490 F TTC
STAR LC 10 COULEUR	1590 F TTC
STAR LC 2410	2090 F TTC
CITIZEN SWIFT 24	2450 F TTC
NEC P2 +	2580 F TTC

14


13


12


11


10

GENERAL

PARIS 10e
Tél. 42.06.50.50
LYON 1er
Tél. 72.00.96.96

LE GRAND SPECIALISTE ATARI

PERIPHERIQUES ATARI

NOS PRIX SONT ANNONCES TOUTES TAXES COMPRISES

TABLES TRAÇANTES

ROLAND DXY 1100 (ft A3) 11620F
ROLAND DXY 1200 (ft A3) 16485F
ROLAND DXY 1300 (ft A3) 22170F

LOGICIELS D'ACCOMPAGNEMENT POUR TABLES TRAÇANTES ROLAND

DYNACADE 7990F
ZZ DRAFT 395F
ZZ BIRD 2D mécanique 948F
ZZ BIRD 2D bâtiment 948F
ZZ BIRD 2D hydraulique 948F
ZZ CONVERT DXF 948F
ZZ CONVERT PLOTTER 948F
ZZ CONVERT ASCII 948F
ZZ ROUGH VERSION 1.1 .... 395F
ZZ VOLUME 11180F

TABLETTES GRAPHIQUES

TABLETTE GRAPHIQUE CRP4 FORMAT A4 .... 4490F
TABLETTE GRAPHIQUE CRP3 FORMAT A3 .... 8490F
TAB. GRAP. SUMMASKETCH A5 .... 4950F HT
TAB. GRAP. SUMMASKETCH A4 .... 6450F HT
TAB. GRAP. SUMMASKETCH A3 .... 9250F HT
KIT ATARI (soft + interface) POUR SUMMASKETCH .... 500F HT

SCANNERS

pour ATARI 520/1040 STE MEGA ST 2 et MEGA ST 4

CAMERON PERSONAL A4 .... 6990F
IMPRIMANTE/SCANNER/PHOTOCOPIEUR.
HANDY SCANNER TYPE 10 .... 2990F
SCANNER CLAVIUS 1490F
SCANNER CANON IX-12 (Type à rouleau) + ZZ SCAN 10900F
SCANNER CANON IX-12F (Type à plat) + ZZ SCAN 16000F

MANETTES DE JEUX

QUICK SHOT 1 .... 69F
Joystick économique, fabriqué par SPECTRAVIDEO. Type à frottement. Résistance moyenne.
QUICK SHOT 2 .... 89F
Joystick à frottement de chez SPECTRAVIDEO. Superbe poignée.
TURBO 6 .... 139F
Joystick à micro contacts, 6 directions, de SPECTRAVIDEO. Le premier prix pour les micro contacts.
JOYSTICK KONIX 149F
Forme très ergonomique. Micro contacts. Moyennement précis.
JOYSTICK PRO 500 .... 129F
JET FIGHTER 225F
Microswitch. Tir automatique.
JOYSTICK COBRA 495F
Le monstre. 30 cm de haut. Une manette de Mirage. La folie complète. Absolument génial, à essayer d'urgence chez GENERAL.
JOYSTICK VOLANT 395F

DISQUES DURS

Cadeau exceptionnel pour tout achat d'un disque dur : un livre Micro-Application "DISQUETTES ET DISQUES DURS" (valeur 179 F)

MEGA FILE 30 ATARI 4190F
Disque dur 30 Mo formaté pour 520 STF, 1040 STF et MEGA ST.
MEGA FILE 60 ATARI 7665F
Disque dur 60 Mo formaté pour 520 STF, 1040 STF et MEGA ST.
MEGA FILE 44 (AMOVIBLE) 8490F
CARTOUCHE 44 MEGA 990F

PERIPHERIQUES VIDEO

LES GENLOCKS INSTITUTIONNELS

GST GOLD SP 5390F
Genlock incrustateur multistandard. Filtre électronique pour digitaliseur. Réalisant les fonctions de décodeur, codeur, correcteur vidéo PAL, SECAM et Y-C. Le GST-GOLD SP est conçu pour synchroniser un ordinateur sur une source vidéo, avec incrustateur et codeur PAL et Y-C. Il dispose d'un filtre électronique pour l'utilisation directe d'un digitaliseur couleur.

GST GOLD VP 6490F
Genlock incrustateur multistandard avec vidéo processeur. Réalisant les fonctions de décodeur, correcteur PAL, SECAM et Y-C. Il est conçu pour synchroniser un ordinateur sur une source vidéo avec incrustateur et codeur PAL et Y-C. Il dispose d'un correcteur vidéo pour l'ensemble des paramètres y compris la balance des blancs.

MX GOLD 2490F
Fader, mixeur son pour GST GOLD SP et GST GOLD VP. Pupitre réalisant le FADE IN OUT ainsi que le mixage son des deux sources vidéo pour le GST GOLD SP et le GST GOLD VP.

LES GENLOCKS PROFESSIONNELS

GST GOLD PRO 6590F
Genlock codeur incrustateur PAL. C'est un matériel professionnel broadcast. Il offre une résolution de 625 lignes et s'adapte à toutes les régies vidéo PAL. Il dispose d'un codeur PAL avec asservissement du color-subcarrier sur le signal vidéo in ou blackburst. Modes d'incrustation : incrustation zéro détecté, incrustation directe ou inverse, chroma key input. Boîtier de télécommande pour FADE IN et OUT.

GST GOLD PRO Y.C 6850F
Genlock codeur incrustateur PAL et Y.C. C'est un matériel professionnel broadcast. Il offre une résolution de 625 lignes entrelacées et 500 points par ligne. Il s'adapte à toutes les régies vidéo PAL, il dispose d'un codeur PAL avec asservissement du color-subcarrier sur le signal VIDEO IN ou blackburst. Modes d'incrustation : incrustation zéro détecté, incrustation directe ou inverse, chroma key input, key output. Boîtier de télécommande pour FADE IN et OUT.

GST 2500

GST 2500 .... 16750F
Atari ou Amiga, un seul genlock. Genlock, décodeur, codeur, incrustateur polystandard. Entrées et sorties : PAL, RVB, Y.C, BETA composantes. Synchronise un ordinateur Atari ou Amiga soit sur une source vidéo de référence PAL, RVB Y-C ou composantes, soit sur un générateur interne de blackburst normalisé. Le GST 2500 est un outil vidéo performant de qualité broadcast permettant non seulement le codage d'un ordinateur vidéo mais aussi le transfert de standard, exemple : PAL ou Y.C en RVB ou Composante BETA. Il s'adapte à toutes les régies PAL, Y.C, ou BETA.

CAMERA NOIR ET BLANC PANASONIC WV1410 .... 2995F
Caméra haute résolution (livrée sans objectif). Faible seuil d'éclairage avec objectif 1.4, monture C. Nombreux objectifs VIVITAR, dont zoom 6x avec position macro, disponibles en option. Nous consulter.

CAMERA NOIR ET BLANC MONACOR TVC 500 .... 1995F

BANC DE REPRODUCTION RB3 KAISER 1290F
Plateau noir mat anti-reflet. Colonnes du banc pourvues d'une échelle graduée. Réglage en hauteur par manivelle. Plateau 400x420x25 mm. Hauteur colonne : 760 mm.

DISPOSITIF D'ECLAIRAGE RB3 KAISER 500F
Convient pour banc de reproduction RB3. Avec 2 douilles de lampe, coulissables et orientables.

PROMOTION
BANC DE REPRODUCTION KAISER RB3 + DISPOSITIF D'ECLAIRAGE RB3 + CAMERA N/B MONACOR AVEC OBJECTIF 3590F

EXTENSION MEMOIRE

512 Ko pour 520 STF 1290F
512 Ko pour 520 STE 990F

RANGEMENT ET ENTRETIEN

BOITES DE RANGEMENT 3"1/2

DISC FILE 5 ALLSOP 3"1/2 .... 39 F
Coffret en plastique injecté avec présentation des 5 disquettes lors de l'ouverture du coffret. Colori crème.

DISC FILE 10 ALLSOP 3"1/2 .... 49 F
Coffret en plastique injecté avec présentation des 10 disquettes lors de l'ouverture du coffret. Coloris : crème.

COFFRET 40 DISKS 3"1/2 DD40L .... 139 F
Coffret pour 40 disquettes 3"1/2 avec capot transparent fumé sur charnière, serrure à clef. Intercalaires pour disquettes fournis. Coloris : noir.

COFFRET 80 DISKS 3"1/2 DD80L .... 175 F
Coffret pour 80 disquettes 3"1/2 avec capot transparent fumé sur charnières, serrure à clef. Intercalaires fournis. Coloris : noir.

MEDIA BOS POSSO 150 DISKS 3"1/2 .... 139 F
Coffret modulaire de fabrication française. Tiroir à disquettes avec intercalaires. Coloris : gris. Serrure en option : 50 F.

COFFRET DD40 DISQUETTES 3"1/2 .... 99 F
Coffret pour 40 disquettes 3"1/2 avec boîtier transparent fumé sur charnières. Intercalaires fournis.

HOUSSES

Housses souples en matière anti-statique. Couleur noire avec liseret rouge.

CLAVIER 520 STF/1040 STF 80F
CLAVIER MEGA ST 2/MEGA ST 4 .... 75F
MONITEUR MONO. SM 124 .... 80F
MONITEUR COULEUR SC 1425 .... 95F
UNITE CENTRALE MEGA ST 2/4 .... 80F
DISQUE DUR SH 205 .... 80F
TOUS MODELES IMPRIMANTES 80 COLONNES 80F
TOUS MODELES IMPRIMANTES 132 COLONNES 95F
IMPRIMANTE LASER SLM 804 .... 150F

BOMBES AEROSOL

COMPUNETT 99F
PRINTER 66 .... 95F
DISQUETTE DE NETTOYAGE 3"1/2 AVEC SON LIQUIDE 99F

MONITEURS

ATARI MONOCHROME SM 124 .... 1490F
ZZ SCREEN MP 19 pouces 22470F

SON

ST REPLAY 4.0 (pour 520 STF ou 1040 STF) 670F
Cartouche + programme. Création de bruits à l'aide d'un micro. Enregistrement à partir de la source audio. Trucages possibles grâce à des modifications de l'enregistrement.

PRO SOUND DESIGNER (pour 520 STF ou 1040 STF) 620F
Cartouche avec logiciel. 2 connecteurs RCA permettent de se brancher sur une source hi-fi. Transforme les signaux sons analogiques en numériques.

EMULATEURS

SPECTRE CGR 3800F
SUPER CHARGEUR 2890F

LECTEURS DISKS

MITSUBHISI INTERNE 3P1/2 .... 1290F
A intégrer dans votre ST. Double tête. 720 Ko.
LECTEUR ATARI 3 P 1/2 .... 990F
LECTEUR ATARI 5 P 1/4 .... 1490F

TELEMATIQUE

CABLE DETECTEUR DE SONNERIE POUR REPTEASER 190F
CABLE MINITEL POUR VIDEOTEASER / REPTEASER 150F
(Sub femelle sortie 25 points à DIN 5 broches)
VIDEOTEASER + REPTEASER + CABLE DETECTEUR SONNERIE + CABLE MINITEL POUR VIDEOTEASER 850F

SOURIS / TRACKBALL

SOURIS HANDYMOUSE 399F
Souris garantie 2 ans, pièces et main d'œuvre. De chez CAMERON. Précision et vitesse d'utilisation exceptionnelles. Très ergonomique. 100 % compatible ATARI. Fabrication japonaise.
SOURIS ST 290F
TRACK BALL 390F

DIVERS

COPIEUR MULTIFACE STF 690F
MOUSE MATE 85F
Tapis souris en néoprène. Favorise considérablement la rotation de la boule.

RUBANS IMPRIMANTES
Prix spéciaux pour possesseurs ATARI ST 520 STF, 1040 STF et MEGA ST

RAMES PAPIER

Rame papier non zoné Format A4, bandes caroll détachables 500 feuilles 69F
Rame papier non zoné Format A3, bandes caroll détachables 500 feuilles 99F
Rame Etiquettes 12x3cm le 1000 sur paravent bandes caroll 85F

CABLES DE LIAISON

Quadrupleur de joysticks 145F
Cable Péritel Atari ST vers moniteur ou télévision (2m) 195F
Cable rallonge joystick/souris 20cm 60F
Cable rallonge joystick/souris 2m 95F
Boitier inverseur vidéo pour brancher moniteur mono et couleur et switcher de l'un sur l'autre 250F
Cable son HIFI, jack 3,5 mono et 2 RCA, 2m 90F
Cable imprimante parallèle Centronics .... de 125 à 250F
Cable minitel 165F
Cable Midi pour relier votre ST à l'expandeur 1,20m 60F
Cable extension disk 2m 195F
MONITEUR MASTER 490F
MOUSE MASTER 290F
KIT TONER POUR LASER ATARI SLM 804 .... 790F
KIT TAMBOUR POUR ATARI SLM 804 .... 2965F
FILTRE ECRAN 14 POUCES 195F
Filtre tissé micromailles. Améliore la persistance. Evite le clignotement.
SUPPORT ECRAN ORIENTABLE 230F
Support écran orientable à 360° de latitude et 20° de longitude. Très pratique.
SUPPORT IMPRIMANTE 80 col. 139FF
Support universel pour tout type d'imprimante.

DISQUETTES VIERGES

PAR BOITES DE 10 PIECES (minimum) POUR LA VENTE PAR CORRESPONDANCE

DISQUETTE 3"1/2 BULK 4,90F
DISQUETTE 3"1/2 DEMARQUEES 5,90F
DISQUETTE 3"1/2 SELECT 7,90F
DISQUETTE 3"1/2 FUJI MF 1 DD 8,90F
DISQUETTE 3"1/2 FUJI MF 2 DD 14,90F
DISQUETTE 3"1/2 GOLDSTAR 10,90F
DISQUETTE 3"1/2 MAXELL MF1 .... 10,90F
DISQUETTE 3"1/2 MAXELL MF2 .... 17,00F
DISQUETTE 3"1/2 MAXELL MF 2 HD 37,90F
DISQUETTE 3"1/2 FUJI MF 2 HD 37,90F
DISQUETTE 3"1/2 SENTINEL DF HD 32,90F
DISQUETTE 3"1/2 MAGTRON préformatée 7,90F

PARIS 10e
Tél. 42.06.50.50
LYON 1er
Tél. 72.00.96.96

LE GRAND SPECIALISTE ATARI

DEPARTEMENT MEGASOFTS ATARI ST

En faisant l'acquisition d'un MEGASOFT chez GENERAL, vous avez droit :

- **GARANTIE DU MEILLEUR PRIX :** si vous trouvez moins cher ailleurs avant votre achat, dites-le nous et nous vous proposerons un prix encore meilleur. Si, dans un délai de quinze jours après votre achat, vous trouvez votre soft moins cher ailleurs, nous vous remboursons la différence.
- **GARANTIE DE BON FONCTIONNEMENT :** en cas de défectuosité du logiciel pendant les 15 premiers jours d'utilisation, nous vous l'échangeons.
- **GARANTIE DE FRAICHEUR :** nous ne vous proposons que les dernières versions commercialisées des logiciels.
- **GARANTIE DE CHOIX :** nous nous engageons à vous offrir le plus grand choix de logiciels possible. Si vous ne trouvez pas dans nos annonces le soft désiré, consultez-nous.
- **DROIT A LA DEMO :** si vous le souhaitez et que les conditions s'y prêtent, vous pouvez demander une démonstration du mégasoft que vous souhaitez acquérir.

MEGA COMPILATIONS

ACTION D'ENFER	249 F
ALL TIME FAVORITE	225 F
AMERICAN DREAMS	259 F
ARCADE HITS	249 F
COMPILATION 1	249 F
EUROPEAN DREAMS	259 F
GEN D'OR	249 F
GIANTS	290 F
JUSTICIERS	239 F
MAGNUM 4	290 F
PRECIOUS METAL	249 F
PREMIERE COLLECTION 3	295 F
PREMIERE COLLECTION	249 F
SILVER COLLECTION	279 F
SIMULATION HITS	249 F
STAR VOL. 1	199 F
STORY SO FAR VOL. 1	199 F
STORY SO FAR VOL. 2	199 F
STORY SO FAR VOL. 3	199 F
TRIAD 2	249 F
TRIAD 3	295 F
VAINQUEURS	299 F

NEWS / A PARAITRE

AN HEADS	N.C.
ACTION CONCEPT	N.C.
BAD BLOOD	N.C.
BATTLE OF BRITAIN	245 F
BATTLE MASTER	245 F
BEVERLY HILLS	195 F
BLADE WARRIOR	245 F
BLOCK OUT	245 F
BLUE ANGEL	245 F
CABAL	199 F
COLDITZ	N.C.
COMBO RACER	195 F
DRAGON SLAYER	N.C.
DRAGON STRIKE	N.C.
DRAGON FLIGHT	245 F
DRAVING FORCE	265 F
DYNASTY WAR	245 F
EPIC	N.C.
ESCAPE FROM PLANET	195 F
FIRE FORGET 2	260 F
FIRE AND BRINGSTONE	245 F
F 19	285 F
FLIMBOS QUEST	N.C.
FLOOD	245 F
GHOST AND GOBLIN'S	195 F
HAMERFIRST	249 F
IMPERIUM	245 F
INTERNATIONAL 3D TENNIS	249 F
KICK OFF 2	N.C.
KILLING GAME SHOW	245 F
LAST NINJA 2	245 F
LOST PATROL	245 F
MATRIX MARAUDER	245 F
MIGHT AND MAGIC	N.C.
MIDNIGHT RESISTANCE	N.C.
NUCLEAR WAR	N.C.
OPERATION STEALTH	295 F
POP UP	199 F
POWER BOAT	245 F
PROPHECY	N.C.
ROTOX	N.C.
RAILROAD TYCOON	295 F
RAIDERS	N.C.
ROBOCOP 2	N.C.
SECRET AGENT	195 F
SHADOW OF THE BEAST	349 F
SHADOW WARRIOR	195 F
SKID MARKS	N.C.
STARBLADE	245 F
TEAM YANKEE	N.C.
THUNDER STRIKE	N.C.
TREASURE TRAP	245 F
TIE BREAK	N.C.
TURRICAN	229 F
UNREAL	N.C.
VENUS	195 F
VIKINGS	N.C.
VROOM	N.C.
WARHEAD	245 F
WIPE OUT	N.C.
WORLD CUP SOCCER	195 F
WONDERLAND	N.C.
4D BOXING	N.C.

ARCADE

AFTER THE WAR	199 F
ANTAGO	179 F
BATMAN THE MOVIE	199 F
BLACK TIGER	199 F
BUBBLE +	249 F
CASTLE MASTER	245 F
CHASE HQ	199 F
CRACK DOWN	195 F
CYBERBALL	195 F
DANDARE 3	195 F
DARK CENTURY	235 F
DOUBLE DRAGON 2	199 F
DYNAMITE DUX	199 F
EAGLES RIDER	245 F
EMOTION	195 F
EXTASE	299 F
GHOSTBUSTER 2	239 F
HARICANA	245 F
HEAVY METAL	195 F
HOT ROD	245 F
IMMPOSAMOLE	195 F
INCORRUPTIBLES	199 F
INDY ARCADE	199 F
INFESTATION	N.C.
INTRUDER	235 F
IVANOHE	245 F
JUMPING JACKSON	199 F
KID GLOVES	245 F
KLAX	195 F
KNIGHT FORCE	269 F
LODE RUNNER	195 F
MANIAC MANSION	249 F
NINJA SPIRIT	245 F
NINJA WARRIOR	199 F
P47	249 F
PINBALL MAGIC	199 F
PIPEMANIA	265 F
POWER DRIFT	239 F
PROJECTYLE	195 F
RAINBOW ISLAND	199 F
RESOLUTION 101	235 F
RICK DANGEROUS	259 F
SHERMAN M4	249 F
SONIC BOOM	249 F
SPACE HARRIER 2	249 F
STRIDER	249 F
STRYX	195 F
TINTIN SUR LA LUNE	220 F
TOYOTTES	199 F
TWINWOLRD	235 F
WILD STREET	269 F
X OUT	199 F

AVENTURE

BAT	349 F
BLOODWHYCH	245 F
CHAOS STRIKE BACK	229 F
COLORADO	245 F
DRAGON'S BREATH	295 F
DRAGON'S LAIR	430 F
DRAKKHEN	299 F
DUNGEON MASTER	245 F
EXPLORA 2	329 F
FETICHE MAYA	249 F
FULL METAL PLANETE	249 F
INDY AVENTURE	249 F
KHALAAN	260 F
MANOIR MORTEVILLE	169 F
MAUPITY ISLAND	289 F
MEURTRES A VENISE	240 F
MIDWINTER	260 F
ROCK STAR	219 F
TOWER OF BABEL	275 F
VOYAGEURS DU TEMPS	249 F

PILOTAGE

BLUE ANGEL	245 F
BOMBER	290 F
CONTINENTAL CIRCUS	229 F
F 16	229 F
F 29	245 F
FALCON	299 F
FERRARI FORMULA ONE	245 F
GRAVITY	245 F
GUNSHIP	239 F
SILENT SERVICE	215 F
STUNT CAR	229 F
SUPER CARS	199 F
TURBO OUT RUN	249 F

SOCIETE

BATTLE CHESS	249 F
DAMES SIMULATOR	239 F
SCRABBLE DELUXE	219 F
TRIVIAL PURSUIT	245 F

SPORT

BEACH VOLLEY	229 F
GAZZA SOCCER	269 F
GREAT COURT	249 F
SOCCER	245 F
ITALY 90	195 F
JACK NIKLAUS GOLF	249 F
KICK OFF EXTRA TIME	149 F
PLAYER MANAGER	269 F
TENNIS CUP	249 F
ULTIMATE GOLF	195 F

STRATEGIE

AUSTERLITZ	259 F
BORODINO	259 F
CONFLIT EUROPE	249 F
NORTH AND SOUTH	249 F
POPULOUS	249 F
WATERLOO	259 F
SIM CITY	280 F

TIR

WESTPHASER	299 F

PARIS 10e
Tél. 42.06.50.50
LYON 1er
Tél. 72.00.96.96

LE GRAND SPECIALISTE ATARI

LOGICIELS GRAPHIQUES

CREATOR	990 F
CYBER CAD vers. 1.0	295 F
CYBER CONTROL	595 F
CYBER PAINT 2	695 F
CYBER SCULT	895 F
CYBER STUDIO	800 F
CYBER TEXTURE	595 F
CYBER UNISPEC	595 F
DEGAS ELITE	219 F
DESSINATEUR	590 F
PRINT MASTER +	440 F
SPECTRUM 512	595 F
STAD	800 F
ZZ DRAFT	395 F
ZZ FUN FACE	285 F
ZZ LAZY PAINT	995 F
ZZ ROUGHT	390 F
DIDOT	690 F
DALI 3	590 F
ARABESQUE	990 F

TRAITEMENTS DE TEXTE

1st WORD +	990 F
BECKER TEXT 2	750 F
EVOLUTION	1390 F
FONTZ !	299 F
REDACTEUR	590 F
SCRIPT	790 F
SIGNUM 2	1450 F
TEXTOMAT	390 F
WORDUP	699 F
CALIGRAPHEUR	1490 F
GRAAL TEXT	790 F

TABLEURS + BASES de DONNEES

BECKER CALC (tableur + graoheur)	990 F
CALCOMAT II	595 F
DATAMAT	390 F
MALETTE BUREAUTIQUE (LDW POWER + EMULCOM + ADIMENS)	1500 F
PACK BUREAUTIQUE 520	590 F
PACK BUREAUTIQUE 1040	1290 F
SUPERBASE	595 F
SUPERBASE PRO	1490 F
SUPERBASE PRO 3.0	2490 F
GRAAL BASE	1590 F
GRAAL CALC	790 F

MISE EN PAGE

PUBLISHINGH PARTNER JUNIOR	990 F
PUBLISHING PARTNER MASTER	2490 F
TIMEWORKS PUBLISHER	990 F
CALAMUS	2490 F

LANGAGES

BASIC OMICKRON	990 F
DEVPACK 2	L750 F
GFA ASSEMBLEUR	590 F
INTERPRETEUR C vers. 2.0	595 F
PACK GFA 3.0 (GFA ASSEMBLEUR)	699 F
MISE A JOUR INTERPRETEUR + COMPILATEUR GFA BASIC 3.5	290 F
LATTICE C	1290 F
MARC WILLIAM'S C	1500 F
OSS PASCAL	800 F
STOS COMPILER	245 F
STOS GAME CREATOR FR	459 F
STOS MAESTRO	299 F
STOS SPRITE 600	199 F
WERKS	360 F

COMPTABILITE

GEST INTEGRALE	2320 F
GESTOCK 90	1779 F
LE COMPTABLE 2	790 F
LE GESTIONNAIRE	590 F
MODULE VENTE	1127 F

LOGICIELS MUSIQUE

LES SEQUENCEURS

BIG BANG	1490 F
CREATOR	2360 F
CUBASE	4000 F
NOTATOR	3990 F
PRO 12	690 F
PRO 24 III	2650 F
STUDIO 24	1150 F
TRACK 24	590 F
MIDI JAZZ	390 F
TRANSTAB	390 F
JAZZ BACK	690 F
BIG BOSS	530 F

LES EDITEURS DE PARTITIONS

MASTER SCORE	2800 F
MUSIGRAPH	1200 F
PROSCORE	1550 F

LES EDITEURS DE SON

FM MELODY MAKER	890 F
MUSIC CONSTRUCTION SET	290 F
ST REPLAY 4	670 F
ST REPLAY PRO	1400 F

EMULATEURS

PC SPEED	2490 F
SPECTRE GCR	3800 F
SUPER CHARGEUR	2890 F

GESTION FAMILIALE

GESTCOMPTE	290 F
GESTION BUDGET PERSONNEL (Gest. bancaire + Graphiques + Buget familial permettant de réaliser votre comptabilité personnelle. Gère 10 comptes, 20 postes, 10 CB + nombreuses options).	300 F

DIVERS

TURBO ST	390 F
MULTIDESK ST	295 F
REVOLVER ST	490 F
G+ ST	350 F
CRAFT ST	690 F
DACTYTLE ST	350 F
SPACK ST (Générateur applications graphiques et sonores)	450 F

UTILITAIRES

SRITE ANIMATOR	390 F
DISCOSCOPIE	490 F
ZZ COM	495 F
GRAAL XPER	1190 F
GRAAL GRAF	390 F

EDUCATIFS

CARRAZ

L'OREILLE EN POINTE	N.C.*

L'AXE DES 3-8 ANS (EVEIL)

ASSOCIE (MULTILANGUES)	249 F
BAMBINO FAIT UN PUZZLE	199 F
BAMBINOURS FAIT UN PUZZLE (MULTIL.)	249 F
BAMBINOURS FORMES & COULEURS	220 F
BARRE L'INTRUS (MULTILANGUES)	249 F
BASE ORTHOGRAPHE/LA DICTEE CE1/CE2	249 F
COLLECTION "JE"/5 LOGICIELS	349 F
LES 1001 VOYAGES	290 F
LE PETIT LECTEUR	290 F
LES PETITS COLORIAGES MALINS 1 & 2	199 F
MELODIK (MEM. MUSICALE 4-10 ANS)	199 F
MEMORISE (MULTILANGUES)	249 F
MIRELA (INIT. SOLFEGE 3-8 ANS)	199 F
RECONNAIS-MOI	199 F
RETROUVES L'HISTOIRE	199 F
RODY ET MASTICO	175 F
TRICAROND (4-10 ANS)	199 F

L'AXE DES 8-12 ANS (EVEIL)

BASE ORTHOGRAPHE/LA DICTEE CM1/CM2	249 F
TROUBADOURS	229 F

L'AXE CULTUREL ET SCIENTIFIQUE

AUX ORIGINES DE LA VIE	199 F
VIE ET MORT DES DINOSAURES	220 F

L'AXE LINGUISTIQUE

D.E. CONCORDANCE DES TPS VOL. 2/5e	199 F
D.E. CONCORDANCE DES TPS VOL. 2/4e	199 F
D.E. CONCORDANCE DES TPS VOL. 2/3e	199 F
D.E. CONCORDANCE DES TPS VOL. 2/2nde	199 F
D.E. CONCORDANCE DES TPS VOL. 2/BTS	199 F

L'AXE LITTERAIRE

AU TEMPS JADIS	199 F
IL ETAIT UNE FOIS	220 F
LE TEMPS D'UNE HISTOIRE	220 F

NATHAN/COKTEL

FRANÇAIS PRIMAIRE

APPRENDS-MOI A LIRE 1	245 F
APPRENDS-MOI A LIRE 2	245 F
LANGUE FRANÇAISE CE1	245 F
LANGUE FRANÇAISE CE2	245 F
LANGUE FRANÇAISE CM1	245 F
LANGUE FRANÇAISE CM2	245 F
LES TROIS PETITS COCHONS - 4/6 ans	220 F
LES CASTORS JUNIORS - 9/11 ans	220 F
PICSOU CHASSEUR DE TRESOR 8/10 ans - 11/12 ans	220 F

FRANÇAIS COLLEGE

LANGUE FRANÇAISE 6e	245 F
LANGUE FRANÇAISE 5e	245 F
LANGUE FRANÇAISE 4e	245 F
LANGUE FRANÇAISE 3e	245 F
ECRIRE SANS FAUTES VOL. 1 - 6e/3e	245 F
ECRIRE SANS FAUTE VOL. 2 6e/3e	245 F
FRANÇAIS XVIe AU XXe SIECLES	220 F
FOLLE LECTURE DE DON QUICHOTTE - 6e	220 F
MONTE CRISTO - 6e/3e	220 F
LABYRINTHE D'ERRARE - 11/14 ans	245 F
LABYRINTHE AUX 100 CALCULS	245 F

MATHEMATIQUES COLLEGE

LA BOSSE DES MATHS V.2.1 - 6e	220 F
LA BOSSE DES MATHS V.2.1. - 5e	220 F
LA BOSSE DES MATHS V.2.1. - 4e	220 F
LA BOSSE DES MATHS V.2.1. - 3e	220 F
MATHS 4e	245 F
MATHS 3e	245 F
LABYRINTHE AUX 1000 CALCULS - 14 ans	245 F

MATHEMATIQUES LYCEE

LA BOSSE DES MATHS 1e	245 F
LA BOSSE DES MATHS 2nde	245 F

LANGUES COLLEGE Anglais - Allemand - Espagnol

BALLADE AU PAYS DE BIG BEN V 3	250 F
ENIGME A OXFORD V 3	250 F
VISA POUR HYDE PARK V 2	250 F
ANGLAIS DEBUTANT	245 F
ANGLAIS CONFIRME	245 F
BALLADE OUTRE-RHIN - 6e/5e	250 F
BALLADE A COLOGNE - 6e/5e	250 F
ENIGME A MUNICH - 4e/3e	250 F
BALLADE A SEVILLE	250 F
ENIGME A MADRID 4e/3e	250 F

LANGUES LYCEE Anglais

THE LIFT	245 F
ANGLAIS PERFECTIONNEMENT 2nde	245 F
ANGLAIS PERFECTIONNEMENT 1re	245 F
ANGLAIS TOP NIVEAU 2nd/1re	245 F

HISTOIRE COLLEGE

AU NOM DE L'HERMINE - 5e	220 F

GEOGRAPHIE COLLEGE

OBJECTIF 1 MONDE - 6e	220 F
OBJECTIF 2 MONDE - 5e	220 F
OBJECTIF FRANCE - 4e/3e	220 F
OBJECTIF FRANCE - 4e/3e	220 F

SCIENCES NATURELLES COLLEGE

LA DECOUVERTE DE LA VIE - 6e/5e	220 F
LA DECOUVERTE DE LA TERRE - 4e/3e	220 F
LA DECOUVERTE DE L'HOMME 4e/3e	220 F

ECONOMIE

CHALLENGE - 1re/term.	295 F

VIE PRATIQUE

ATLAS	230 F
CODE FACILE	230 F
MICRO BAC FRANÇAIS - 1re/term	220 F
MICRO BAC MATHS C & E - 1re/term.	220 F
MICRO BAC MATHS D - 1re/term.	220 F
MICRO BAC PHYSIQUE-CHIMIE - 1re/term.	220 F
MICRO BAC ANGLAIS - 1re/term.	220 F
MICRO BAC ALLEMAND	220 F
MICRO BAC ESPAGNOL - 1re/term.	220 F
MICRO BAC GEOGRAPHIE - 1re/term.	220 F
MICRO BAC HISTOIRE - 1re/term.	220 F
MICRO-BREVET FRANÇAIS	220 F
MICRO-BREVET MATHS - ALGEBRE	220 F
MICRO-BREVET MATHS - GEOMETRIE	220 F
MICRO-BREVET HISTOIRE	220 F
MICRO-BREVET GEOGRAPHIE	220 F

COMPILATIONS

MULTICOURS 6e	245 F
MULTICOURS 5e	245 F
MULTICOURS 4e	245 F
MULTICOURS 3e	245 F

UTILITAIRES

ANGLAIS POUR LE BAC	460 F
ENGLISH FOR BUSINESS	490 F
BALADE AU PAYS DE L'ANGLAIS - 6e/2nde	720 F

GENERATION 5

DESTINATION MATH CE1-CE2	265 F
DESTINATION MATH CM1-CM2	265 F
DESTINATION MATH 6e-5e	265 F

MICRO C

FRANÇAIS CM	245 F
FRANÇAIS SN	245 F
ORTHO CM	245 F
GUILLOTINE	199 F
GRAMMAIRE 6.5	245 F
MATHS EDUCATIF PRIMAIRE	245 F
MATH CE	245 F
MATH CM	245 F
MATH 6	245 F
MATH 5	245 F
MATH 4	245 F
MATH 3	245 F
MATH 2	245 F
MATH 1	245 F

VTA

ORTHOGUS VOL 1	245 F
ORTHOGUS VOL 2	245 F
MATHEX 6e/5e	245 F
MATHEX 4e/3e	245 F

SERVICE VPC

10, bd de Strasbourg 75010 PARIS

6 MODES DE REGLEMENT AU CHOIX :

1) **EN CONTRE-REMBOURSEMENT :** vous ne réglez que lors de la livraison de votre matériel. Prévoir 24 F de frais de contre-remboursement lors d'une expédition P&T et 40 F par un transporteur (envoi de plus de 5 kg ou d'une valeur de plus de 2000 F) en sus du forfait transport. Remplissez le bon de commande ci-dessous. Si vous ne souhaitez pas mutiler votre revue, photocopiez-le ou écrivez nous sur papier libre ou encore, téléphonez nous (demander M. BERNARD au (1) 42.06.50.50, postes 436 ou 441).

2) **PAR CARTE BLEUE :** vous pouvez nous téléphoner et nous commander le matériel désiré, en nous donnant le numéro de votre carte bleue. Vous pouvez également utiliser le bon de commande ci-dessous.

3) **PAR CHEQUE OU CCP :** vous remplissez le bon de commande ci-dessous en joignant le règlement du total de votre commande plus le forfait transport correspondant à l'ordre de GENERAL VIDEO (voir ci-contre).

4) **PAR MANDAT :** vous remplissez le bon de commande ci-dessous en précisant que vous désirez régler par mandat postal ou par mandat-lettre. N'omettez pas de compter le forfait transport (voir ci-dessous). A réception de votre mandat, nous expédions votre commande.

5) **PAR MINITEL : TAPEZ 3615 code GVF,** puis choisissez dans notre catalogue Minitel le ou les articles que vous souhaitez commander. Indiquez-nous votre adresse complète, votre numéro de téléphone et votre numéro de Carte Bleue. Si vous souhaitez des renseignements ou une documentation supplémentaire, vous pouvez nous poser vos questions et nous vous répondrons dans les délais les plus courts. Vous pouvez aussi vous amuser sur notre serveur grâce aux concours permanents et gagner de nombreux lots.

Si vous ne pouvez vous déplacer ce service est fait pour vous

6) **A CREDIT :** si vous souhaitez des renseignements détaillés pour l'achat à crédit, vous pouvez joindre Mlle TANIA, Responsable Crédit, au (1) 42.06.50.50, poste 458. Si tout vous parait clair, vous envoyez le bon de commande ci-dessous en précisant si vous souhaitez verser une partie comptant. Par retour du courrier, nous vous envoyons un dossier de crédit que vous devez nous retourner dûment rempli et signé. Après acceptation de ce dossier, nous vous expédions le matériel. Pièces demandées : pièce d'identité, dernier bulletin de salaire, RIB, quittance EDF ou de loyer informatisée.

- **FORFAIT TRANSPORT :** 20 F pour 2 softs, 35 F pour 3 softs et plus.
  50 F pour les accessoires et les disquettes vierges.
  60 F pour les imprimantes.
  120 F pour les machines.

(Dom-Tom et étranger, nous consulter).
Ce forfait n'est pas cumulable, c'est-à-dire que si vous commandez un micro-ordinateur plus des logiciels, vous ne payez que le forfait transport machine, donc 120 F. Tous nos matériels sont expédiés en recommandé.

ATTENTION : pour toute commande supérieure à 10.000 F, joindre un acompte de 20 % à la commande

BON DE COMMANDE EXPRES

Je, soussigné, déclare commander à GENERAL les marchandises et fournitures ci-après désignées pour expédition à mon adresse indiquée ci-contre.

Je choisis de régler par : ☐ Contre-remboursement
☐ Carte Bleue* ☐ Chèque/CCP ☐ Mandat ☐ Crédit

* N° de Carte Bleue — Date expir. CB

N° Carte Aurore

NOM Prénom

N° — Rue

Code Postal

Ville

Tél.

DESIGNATION — Pour toute commande de logiciels, précisez le type de la machine et le format de(s) disquette(s) désiré(es)	QUANTITE	PRIX UNITAIRE	MONTANTS

BON DE COMMANDE A RETOURNER A GENERAL, 10 BD DE STRASBOURG, 75010 PARIS

Avez-vous déjà commandé par correspondance chez GENERAL depuis le 1/02/1989 OUI ☐ NON ☐

Date : .....

Signature

Pour les mineurs, la signature des parents est obligatoire.

Observations du client : .....

	TOTAL COMMANDE	
+	FORFAIT DE PORT	
+	FRAIS CONTRE-REMBOURS.	
	TOTAL A REGLER	

VENTE EN GROS

PLUS DE 6500 ENTREPRISES ET COLLECTIVITES NOUS ONT DEJA FAIT CONFIANCE : VOUS AUSSI, OUVREZ UN COMPTE CHEZ GENERAL

Vous aurez droit à des prix professionnels, livraison sous 24 heures, réglement sur relevé de factures.

La vente en gros s'adresse aux entreprises, collectivités, administrations, etc...

Les membres de ces collectivités peuvent se rendre individuellement chez GENERAL, munis d'un justificatif. Il se verront remettre une CARTE COLLECTIVITÉ qui leur donnera accès aux prix de gros.

Pour toute information, contactez M. N'GOM, M. ALVAREZ ou Mme LE POULL, tél. 42.06.50.50, télex 214.034, télécopie 42.38.35.60.

Entreprises, nous sommes en mesure de vous proposer les services suivants pour l'acquisition de votre matériel chez GENERAL :

1°) ouverture d'un compte, après la première commande ;
2°) un financement personnalisé : crédit-bail pour tout matériel d'une valeur supérieure à 10.000 F à condition d'avoir au moins deux exercices d'activité. Ce mode de financement vous permet en outre de récupérer la TVA sur les loyers mensuels.
3°) une formation d'initiation gratuite de votre personnel avec chaque machine vendue (durée 1/2 journée) ;
4°) une formation approfondie, soit dans nos locaux, soit auprès d'instituts spécialisés qui ont notre confiance tel le prestigieux ALTITUDE 21 ;
5°) une assistance téléphonique, sans limitation de temps ;
6°) une maintenance sur site au travers de puissants groupes spécialisés tels AMTI ou MIS ;
7°) une tarification étudiée en fonction des quantités souhaitées et de votre statut d'entreprise.

ETUDIANTS

NOTRE TARIF COLLECTIVITE EST AUSSI POUR VOUS

En dehors des prix promotionnels et des actions temporaires GENERAL, vous avez droit à un tarif spécial sur l'ensemble des produits de notre magasin. L'inscription pour obtenir la CARTE ETUDIANT GENERAL est immédiate et gratuite

CONDITIONS DE VENTE : ce catalogue n'est valable que pour le mois de sa parution et remplace les précédentes éditions. Produits proposés dans la limite des stocks disponibles. Nos promotions ne sont pas cumulables

Désormais, la Rubrique Jeux de ST Mag présente ses photos avec un cadre de couleur afin d'identifier la version photographiée :
- cadre rouge : Amiga.
- cadre bleu : ST.

SHADOW WARRIOR

OCEAN **AMIGA**

Je me rappelle qu'il y a quelques mois, je passais des heures devant une machine d'arcade sur laquelle on trouvait un jeu de combat fabuleux nommé Shadow Warrior. Eh bien, ça y est, il est arrivé sur Amiga, du moins pour le moment (je l'espère), et je dois avouer que la version micro est exactement la même que l'arcade. Les graphismes sont identiques et complétés par une animation très fluide (mais plus lente que l'arcade). Vous aurez aussi la joie de pouvoir jouer à deux (en même temps) et grâce à vos talents conjugués, vous aurez plus de facilité à vaincre les chefs de gang. De plus, vous disposez de cinq crédits (au départ) et d'une option "continue" pour vous venir en aide.
Le scénario est des plus simples, puisque votre but est de traverser une ville en "blastant" tout ce qui passe à votre portée (même les poubelles) jusqu'au 6ème niveau où vous attend un démon qui n'est autre que le Big Boss. A mon avis, il aurait mieux fallu faire intervenir l'équipe de Ghostbusters...

Le maniement du personnage est aisé et vous aurez la surprise de le voir changer ses coups en fonction du décor. Parfois, à la place de sauter et de donner un coup, il s'agrippe à un lampadaire et saute sur sa proie à pieds joints. Il sait aussi faire des sauts périlleux qui sont assez pratiques si vous les combinez avec un mur. En effet, en sautant sur le mur, vous vous retrouvez derrière vos ennemis et boum !
En bref, Shadow Warrior est une superbe adaptation fort bien réalisée et c'est avec impatience que nous attendons les autres versions micro.

Note : 18/20 **Jean Delaite**

FLOOD

ELECTRONIC ARTS **AMIGA**

Ça y est, l'équipe de Populous nous sort un nouveau jeu complètement différent puisqu'il s'agit d'un jeu d'arcade du type de Rainbow Island et de Bubble Bobble. Les graphismes (surtout la présentation) et l'animation sont superbes. Un tout qui forme un jeu plein de fun et d'actions.

Flood narre l'histoire de Quiffy, un petit monstre dont le monde est envahi de poubelles, de déchets en tous genres et même de monstres. Mais ce n'est pas tout, car une inondation menace aussi ce monde sous-terrain. Bravant les éléments, Quiffy doit donc nettoyer et sauver son monde de la destruction. Une fois le tableau propre, Quiffy doit retrouver un téléporteur qui l'enverra dans une autre salle. Il y a 42 salles au total, et dans certaines, vous trouverez des codes d'accès qui vous permettront de revenir à ce niveau en cas de "malheurs", mais cherchez bien car ces codes sont bien cachés.

La vie de Quiffy est régie par deux niveaux : un rouge pour l'énergie et un bleu pour la réserve d'air lorsque vous êtes sous l'eau. Au niveau des monstres, vous en avez toute une galerie. En voici donc quelques-uns : les Psycho Teddies sont des nounours en peluche affamés et dotés d'une énorme bouche sur le ventre. Il y a aussi des mousses qui vous tuent au premier contact et puis il y a le fantôme qui suit tous vos mouvements pas à pas et qui est indestructible. Le seul moyen de lui échapper est de terminer le tableau au plus vite. Vous avez aussi des armes à votre disposition, à concurrence d'une par niveau. Elles peuvent aller du boomerang au lance-flammes en passant par les grenades.

En conclusion, en raison du mauvais temps prévu cet été, vous ne pourrez pas sortir à la plage, alors pourquoi ne pas jouer à Flood ?

Note : 18/20 **Jean Delaite**

FALCON MISSION DISK

SPECTRUM HOLOBYTE ST

Enfin le voilà, le nouveau "mission disk" pour mon simulateur de vol préféré. Tout ce que je peux vous dire du scénario, c'est qu'il s'agit d'une grande bataille dans laquelle les missions s'enchaînent les unes aux autres et que la mission suivante dépend de votre action dans la précédente. On peut parler de missions interactives. Les nouveautés, il y en a quelques-unes... La première surprise est pour la page de présentation, en effet cette dernière a changé (je préférais le dessin du faucon dans la première version, m'enfin !). Les pilotes des Migs sont beaucoup plus coriaces à larguer et s'accrochent à vos ailes comme de véritables tiques. L'animation est encore plus fluide et plus rapide (à moins que ce ne soit qu'une impression). Les graphismes aussi ont été soignés et je vous conseille de survoler le convoi de camions ainsi que le train à très très basse altitude, c'est époustouflant !

Décidément, les concepteurs de Falcon ont décidé de le rajeunir petit à petit, ce qui n'est pas pour me déplaire et si comme moi, vous êtes un inconditionnel de ce simulateur de vol, alors ce Mission Disk II est pour vous.

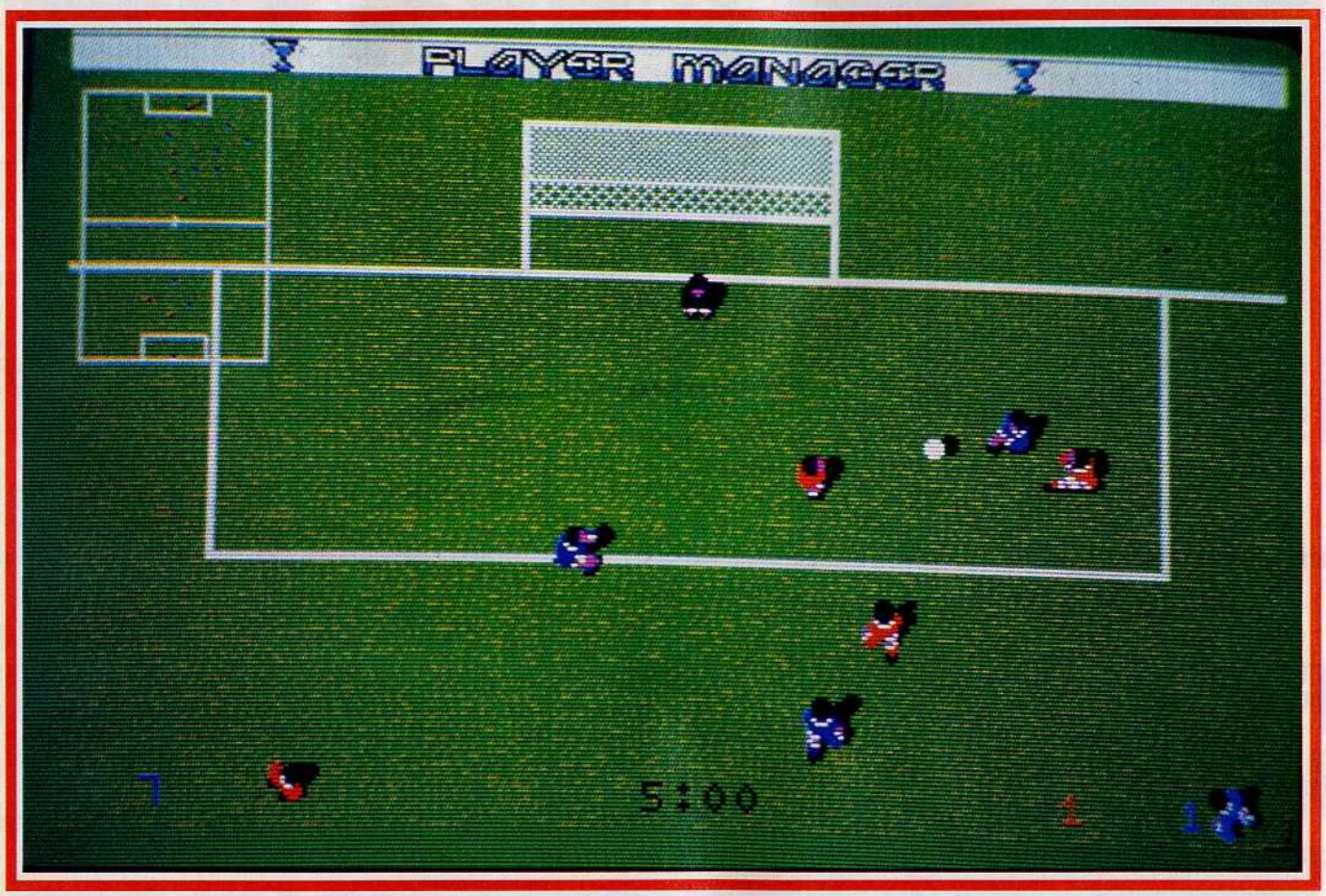
Note : 18/20 **Jean Delaite**

KICK OFF 2

ANCO **AMIGA**

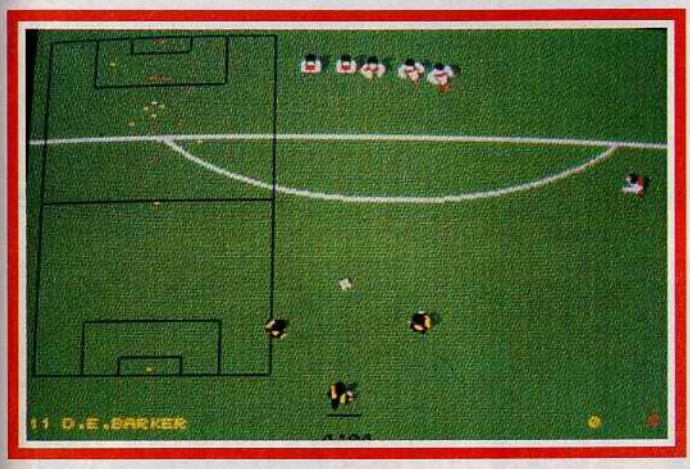
Il y a eu Kick Off (une superbe nouveauté mais plutôt buggée) puis Kick Off Extra Time et enfin Player Manager (qui permettait de créer ses propres équipes). Aujourd'hui, vous faites ce que l'on appelle un remix et vous agitez le tout très fort. Le cocktail se nomme Kick Off 2 (le retour...). Il existe déjà pas mal de simulateurs de football, plus ou moins bons, qui sortent sur le marché, mais aucun ne pourra arriver là où Kick Off 2 nous entraîne.

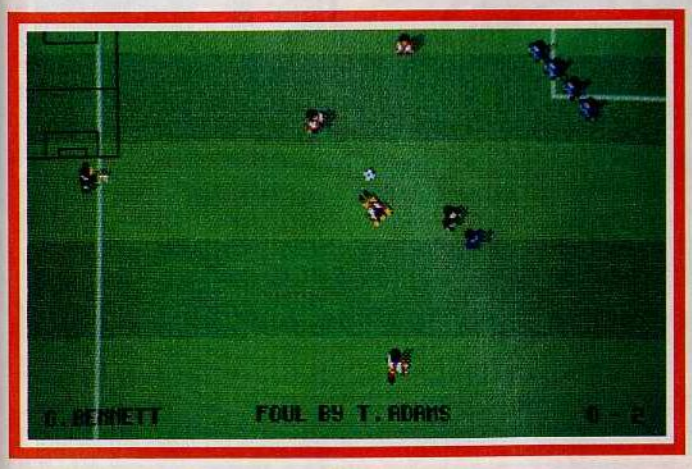
Kick Off 2 possède la même disposition à l'écran que le premier, sans les bugs, et avec de plus beaux graphismes. Chaque équipe est paramétrable (jusqu'à la couleur des maillots). Il existe différents types de terrain : normal, mouillé, artificiel et détrempé. Bien sûr, vous pouvez aussi faire confiance au hasard. Il faut aussi noter que le jeu change en fonction du terrain et du vent (eh oui, il y en a aussi). On peut aussi choisir le temps de jeu. En cas de match nul, on passe aux prolongations et si durant celles-ci, vous n'avez pas résolu votre problème, vous aurez à tirer des pénalties. J'oubliais une chose très importante : on peut choisir son arbitre !...
Kick Off 2 vous propose plusieurs types de match, avec l'entraînement, le match amical, le championnat mais aussi un scénario : la coupe du Monde 90.



Un des "Plus" de Kick Off 2 réside dans le fait que l'on peut y jouer jusqu'à quatre joueurs (avec extension joystick) en 4 équipes ou bien à 4 dans la même équipe. En effet, vous avez la possibilité de ne jouer qu'une place durant tout le match, l'ordinateur gérant les autres joueurs.
Au niveau des déplacements, le goal est capable de faire des tonnes de coups différents lors des arrêts mais aussi lors des dégagements. Les joueurs peuvent aussi brosser les tirs (c'est dur à réaliser, mais superbe pour marquer). Mais là où Kick Off 2 est génial, c'est qu'il existe un système de sauvegarde de vos meilleures actions (et buts) pour vous permettre d'épater la galerie plus tard.

Note : 20/20 **Jean Delaite**





20th CENTURY SOFT

☎ 70.46.20.48

ADV RUGBY SIMULATOR
BLOOD MONEY
BLOODWYCH
DYNAMITE DUX
INDIANA LAST CRUSADE
PASSING SHOT
PIRATES
POPULOUS SCENARY
VIGILANTE
ZAC MAC KRACKEN

POUR 3 JEUX ACHETES
1 JEU GRATUIT
(DANS LA LISTE CI-DESSUS)

DANS LA LIMITE DES STOCKS DISPONIBLES

NOUVEAUTES

ANARCHY	189
BACK TO THE FUTURE2	229
BLOCK OUT	225
BOMBER MISSION DISC.	165
DAMOCLES	249
DEFENDER OF THE EARTH	189
DYNASTY WARS	189
DYTER 07	229
ESCAPE FROM PLANET	165
EXTASE	225
FIRE & FORGET 2	249
FIRE BRIMSTONE	225
FLOOD	225
GOLD OF AMERICAS	225
HAMMER FIST	229
IMPERIUM	225
LAST NINJA 3	225
LOST PATROL	189
ORIENTAL GAMES	225
PROJECTYLE	229
RAIDERS	229
RESOLUTION 101	249
RINGS OF MEDUSA	249
ROBOCS DRIFT	229
ROTOX	189
SECRET AGENT	165
SECRET DEFENSE	275
SHADOW WARRIOR	165
SIM CITY	249
TIE BREAK	229

COMPILATIONS

DUNGEON MASTER + CHAOS STRIKE BACK	299
FALCON + MISSION DISC.	299
ACTION DENIER	249
ALL TIMES FAVORITES	249
AMERICAN DREAMS	245
MAGNUM 4	229
MASTER COLLECTION	299
MEGA PACK 2	299
MICROWORLD MAD A 80s	419
PRECIOUS METAL	225
SILVER COLLECTION	299
TOP HIT	199
TRIAD	225
TRIAD 3	225

PROMO

F19	265
F29	199
FALCON	199
ITALY 90	199
KICK OFF 2	185
DUNGEON MASTER	199

UTILITAIRES

SUPERBASE 2 + LIVRE +	
BECKER CALC + T.SHIRT	990
BECKER TEXT 2	490
CALAMUS	2250
CALCOMAT 2 PLUS	490
DATAMAT	390
LE REDACTEUR 3	690
SUPERCHARGER	2650

1 ST WORLD PLUS	690
A - DEBOG	570
ARABESQUE	250
AUTOFORMAT GFA OU MCo	290
BECKER CAD	990
BIG BAND	1690
BUREAU PERFORMANCE	1250
CALCOMAT PLUS	365
CALLIGRAPHER JUNIOR	250
CALLIGRAPHER PRO	1850
COMPTA 2 MENSOFT	2990
CYBER PAINT	590
CYBER SCULPT	790
DALI S	570
DEVPAC V2	710
DISCOSCOPIE	470
FLEXDISC	250
FM MELODY MAKER	790
FUN FACE	290
FONTZ	270
GESTCOMPTES	290
GESTION INTEGRALE	1990
GESTION BUDGET PER V2	290
GFA ARTIST	490
GFA ASSEMBLEUR	570
GFA BASIC 3.0	650
GFA COMPIL + BASIC 3.5	250
GRAPHIC TOOLBOX	650
HOT WIRE	490
HOUSE MUSIC SYSTEM	650
K SPREAD 2	650
K SPREAD 4	950
LATTICE C	1250
LAZER C	1690
LE COMPTABLE 2	750
LE DESSINATEUR	650
LE GESTIONNAIRE	650
LE REDACTEUR 1.00	650
MAGIC SAC PLUS	990
MARK WILLIAMS S.O.	1290
MASTER SOUND	495
MIDI JAZZ	940
MULTIDESK	940
PACK GFA BASIC 3.0	750
PACK OMIKRON	690
PACK LDW POWER	1650
PRINT MASTER +	350
PRO 12	670
PROTOS	350
PUBLISHING PART JUNIOR	670
PUBLISHING PART MASTER	2990
PUBLISHING PART V 1.0	1590
QUARTET	490
REVOLVER	490
SCRIPT	750
SOLUTION PERSONNELLE	510
SPACK	450
SPECTRE 128	1450
ST REPLAY V4 VF	690
ST REPLAY PACK PRO	1350
STOS BASIC	350
STOS BASIC VERS. FRA	490
STOS COMPILER	345
STOS GAMES GALORE	350
STOS MAESTRO	590
STOS MAESTRO PLUS	690
STOS SPRITE 600	390
STUDIO 24	1690
SUPERBASE	990
SUPERBASE 2	650
SUPERBASE PRO + LIVRE	350
TEXTOMAT	345
TRACK 24	490
TURBO ST V 1.8	350
TWIST 2	490
U.I.S. 2	250
WIDE ST + MIXIMAGE	2150
WERCS	515
ZZ COM	490
ZZ DRAFT	350
ZZ LAZY PAINT	790
ZZ ROUGH	690

AFTER THE WAR	189
ANTAGO	189
AQUANAUT	249
ARCHIPELAGOS	199
ARMADA	269
AUSTERLITZ	279
BAD COMPANY	249
B.A.T.	249
BATTLE CHESS	249
BLACK TIGER	169
BOMBER	295
BORODINO	329
CASTLE MASTER	259
CHAMBERS OF SHAOLIN	215
CHAOS STRIKE BACK	199
CHASE HQ	169
CHESSPLAYER	290
CHICAGO 90	259
CONQUEROR	259
CRACK DOWN	189
CRAZY CARS II	215
DAMES SIMULATOR	259
DARK CENTURY	259
DEJA VU 2	249
DOUBLE DRAGON II	189
DRAGON'S BREATH	279
DRAKKEN	269
DUNGEON MASTER VF	255
E - MOTION	165
E.S.S HERMES	269
EXPLORA 2	279
EXTRA TIME (100K OFF)	119
EYE OF HORUS	259
F 19 COMBAT PILOT	229
FALCON	229
FALCON MISSION 1	195
FERRARI FORMULE 1	225
FIENDISH FREDDY	269
FRED	259
FLIGHT SIMULATOR II VF	345
FOFT	199
FULL METAL PLANETE	250
GHOSTBUSTER 2	229
GHOST GOBELINS	169
GREAT COURTS	245
GUNSHIP	229
HARD DRIVIN	169
HEAVY METAL	179
HEROES OF THE LANCE	255
HOUND OF SHADOW	225
INFOGRAMOLE	169
INDIANA AVENTURE VF	249
INFESTATION	229
INTERPHASE 3D	229
IRON LORD	269
IVANHOE	165
JET + SCENARY. JAPAN	329
JUMPING JACK SON	219

KHALAAN	269
KLAX	165
KNIGHT FORCE	199
LA LEGEND DU DUEL	215
LES VOYAGEURS DU TEM.	259
LES INCORRUPTIBLES	169
MANIAC MANSION VF	259
MANOIR DE MORTEVILLE	195
MAURITI ISLAND	269
MICROPROSE SOCCER	229
MIDWINTER	269
NEVERMIND	165
NEW ZELAND STORY	179
NIL DIEU VIVANT	259
NINJA WARRIOR	189
NORTH AND SOUTH	199
OCEAN BEACH VOLLEY	169
OIL IMPERIUM	199
ONSLAUGHT	169
OPERATION NEPTUNE	259
OPERATION THUNDERBOLT	169
P 47	259
PIPEMANIA	255
PLAYER MANAGER	249
POPULOUS	215
RAINBOW ISLAND	169
RED STORM RISING VF	295
RICK DANGEROUS	169
ROCKET RANGERS	255
ROCK STAR	199
SAFARI GUNS	169
SHINOBI	169
SILENT SERVICE	199
SLEEPING GODS LIE	259
SORCERER LORD	269
STEVE DAVIS SNOOKER	195
STRIDER	169
TENNIS DE TABLE	199
TOWER OF BABEL	249
TRIVIAL PURSUIT JR	279
TV SPORTS FOOTBALL	229
TWIN WORLD	245
ULTIMA V	275
ULTIMATE GOLF	255
VIRUS	189
WATERLOO	259
WILD STREETS	259
XENON II	219

PROMO

DISQUETTES
3" 1/2 DF DD
5.50 F par 100
5.90 F pièce par 50
6.40 F pièce par 10

PORT GRATUIT OU DOMPUB

A FRAIS DE 30 F D'ACHAT DE LOGICIELS

A RETOURNER A : CENTURY SOFT B.P. 454 03004 MOULINS CEDEX
(OU RECOPIER)

NOM : ☎ 70.46.20.48

ADRESSE : ☐ CONTRE REMBOURSEMENT + 24 F

☐ CHEQUE ☐ CARTE BLEUE

VILLE N° CB

CODE POSTAL TELEPHONE DATE D'EXPIRATION

TITRES : PRIX :

FRAIS D'EXPEDITION :
☐ NORMAL 18 F
☐ COLISSIMO 25 F
Livraison garantie sous 48 H
☐ PORT 50 DISCS 55 F
100 DISCS 60 F

FRAIS DE PORT
TOTAL :

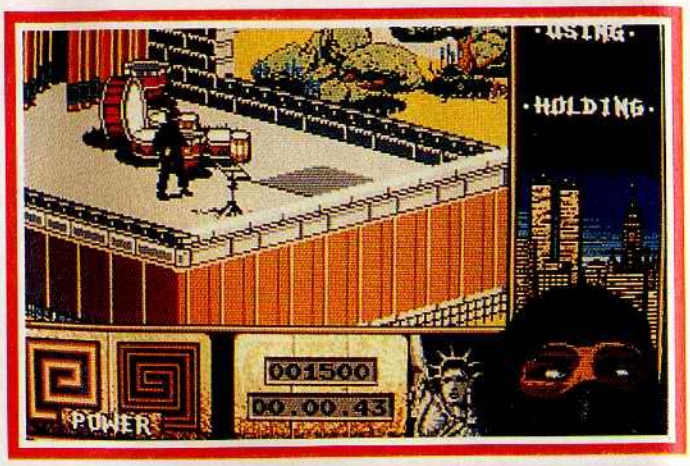
SIGNATURE :

ST MAG 48

LAST NINJA II

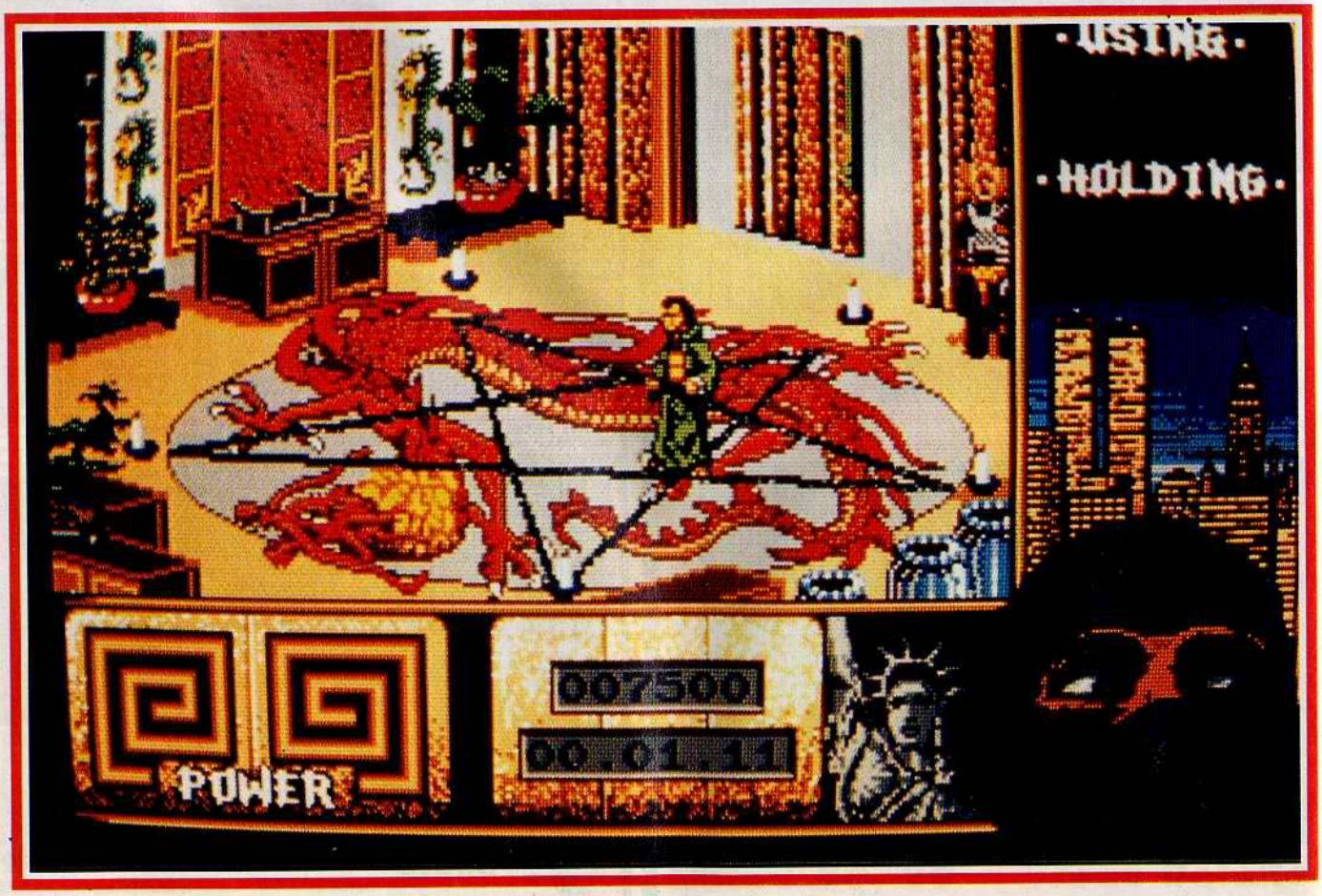
SYSTEME 3 **AMIGA / ST**

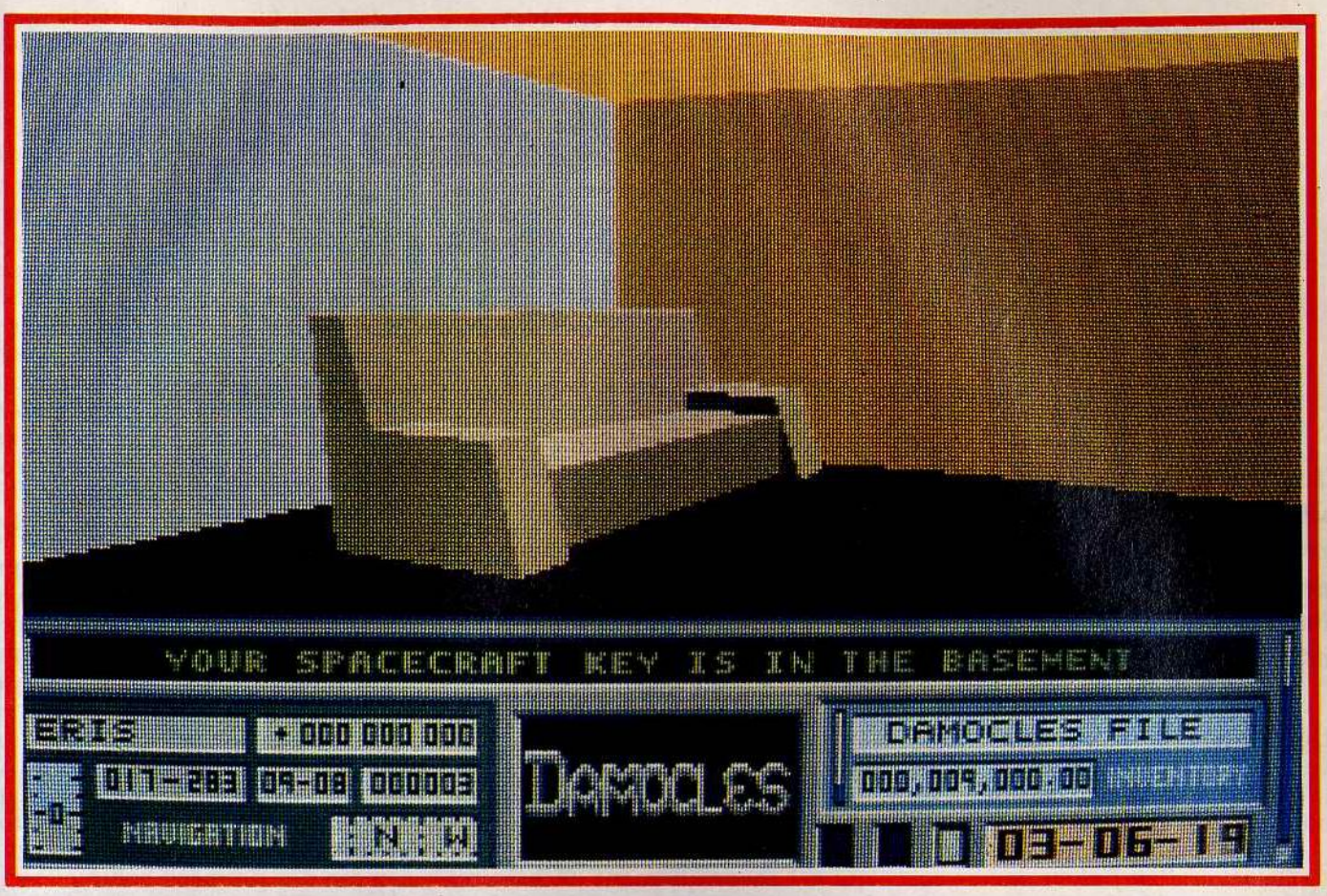
Tiens, un numéro 2 ? Et pour une fois, complètement différent du premier. Last ninja II allie à la fois arcade et aventure (dans le sens le plus simple du terme), le tout emballé dans un décor en 3D du type Krafton & Xunk. Personnellement, je préférais la première version dans laquelle on se défoulait en frappant tout ce qui bouge sans se poser de questions. Du côté des graphismes, c'est bien mais loin d'être exceptionnel. Il y a aussi une autre chose qui m'a vraiment gêné : le Ninja n'a que deux coups possibles. Quand on connaît les autres jeux avec des Ninja, c'est vraiment très peu.



L'ordre des Ninja est mort vers le XIIème siècle, mais des parchemins ont été retrouvés et Armakuni (alias vous), le grand maître, tente de le reformer. Alors qu'il était entouré de ses élèves, une aura mystérieuse l'enrobe et le téléporte dans un monde étrange. C'est ainsi qu'Armakuni se retrouve à New-York en plein XXème siècle. Son but est de retrouver celui qui l'a fait venir ici et de le détruire (quelle prétention). Ce ne sera pas une chose facile car l'autre est vraiment bien protégé par une horde de sbires assoiffés de sang et de tripes (beurk!) (NDLR: petite touche personnelle pour enjoliver l'histoire).
Le jeu est constitué de six niveaux : Central Park, les rues de Manhattan, les égouts (avec plein de monstres bizarres), l'usine d'opium (je préfèrais les tripes), un labyrinthe de bureaux et enfin l'antre de votre ravisseur qui n'est autre que le Shogun Kunitoki, votre ennemi ancestral.
En conclusion, j'émettrai quelques réserves quant à la qualité de ce deuxième volet de Last Ninja en partie à cause des graphismes et de l'animation, mais surtout à cause des commandes qui ne sont pas très pratiques à mon goût.

Note : 12/20 **Jean Delaite**



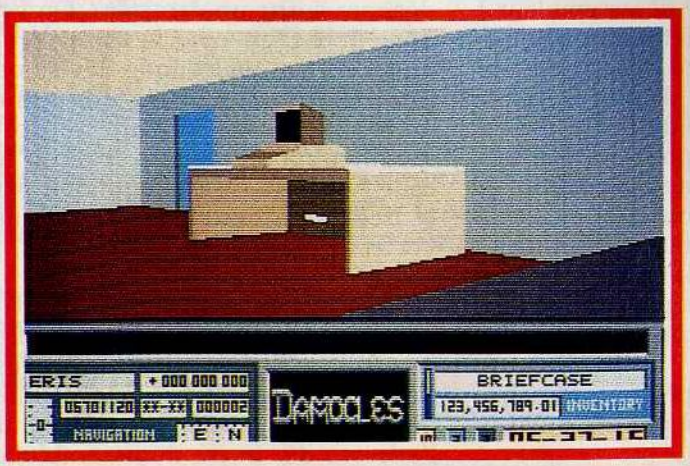


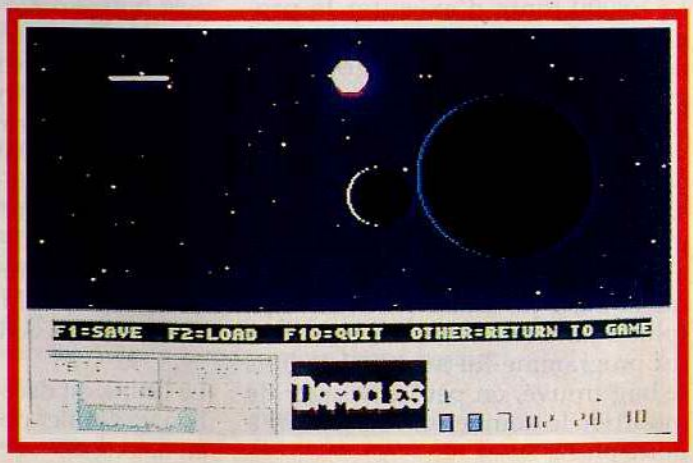
DAMOCLES

NOVAGEN **AMIGA / ST**

Ça fait plus de deux ans que Novagen nous annonce la sortie de la suite de Mercenary. C'est enfin fait.
La planète Eris vit dans l'angoisse permanente d'une collision avec l'astéroïde Damoclès (vous comprenez le nom maintenant). Les savants sont désemparés (comme toujours) et c'est à vous que le gouvernement fait appel. Lors du voyage, vous avez un problème avec votre caisson d'hibernation et, au lieu d'arriver bien avant la catastrophe, vous débarquez trois heures avant sur la planète ! Derrière un jeu d'apparence simpliste se cache un scénario embrouillé et plein de surprises. Le système de jeu est le même que dans Mercenary, avec une meilleure animation et des formes en 3D pleines. Le personnage peut marcher, courir, conduire un véhicule terrestre, aérien et même interplanétaire.
Damoclès pourrait être un bon jeu mais je persiste à penser qu'il aurait été bon de faire intervenir de vrais personnages et non des phrases sur l'écran, afin de le rendre plus captivant. Je vous dis ça parce qu'au bout d'une heure de jeu j'ai commencé à m'ennuyer sérieusement. Damoclès aurait fait un excellent hit il y a deux ans. Dommage.

Note : 12/20 **Jean Delaite**





UN DÉSASSEMBLEUR POUR 68000

(Amiga, mais aussi ST et autres machines à base de 68000).

Tous les programmeurs en assembleur 68000 ont, ou auront à utiliser un jour un désassembleur. En effet, cet utilitaire est pratiquement le seul moyen d'investigation dont dispose le programmeur pour trouver ce que l'on appelle poliment les bugs (on leur trouve souvent des noms plus imagés quand on est en prise avec l'un d'entre eux).

L'utilisation du désassembleur trouve sa place dans l'une des phases de la création d'un programme que nous pouvons détailler précisément :

1- Création d'un fichier source en langage assembleur.
Ce dernier est un langage à base de mnémoniques simples à mémoriser et à traiter. Pour plus d'informations, il faut se reporter aux initiations à l'assembleur de ST MAG ;
2- Assemblage de ce fichier.
Le fichier source est transformé en fichier objet par un programme nommé Assembleur. Le fichier objet contient du code machine qui est compris directement par le 68000. Par contre, il est difficile de se retrouver dans ce code sans l'aide de notre désassembleur.
3- Exécution.
Le 68000 tente d'exécuter le programme mais comme d'habitude il n'a rien compris à l'algorithme génial qu'on a vainement essayé de lui expliquer (arg!) ;
4- Débogage: Exécution en pas à pas avec désassemblage.
Notre utilitaire va ici le devenir ("utile"...), en recréant un listing en assembleur à partir du contenu mémoire, c'est-à-dire du code objet (plus ou moins altéré par l'éxécution du programme lui-même). Une fois le bug trouvé, on peut envisager de modifier le source en retournant à notre phase 1.

Le désassembleur est donc un utilitaire d'aide à la lecture du code machine, mais comme dans toutes les applications où le nombre d'informations disponible est inférieur à celui qui est théoriquement nécessaire, l'ordinateur va être contraint de faire un choix. En effet, le code machine généré par l'assembleur contient moins d'informations que le code source : on perd, lors de l'assemblage, tous les commentaires, les labels (sauf avec l'option EXPORTS mais le désassembleur étudié n'en tient pas compte) et finalement toutes les variables évaluées lors de l'assemblage. De plus le 68000, comme d'ailleurs la majorité des microprocesseurs actuels, ne fait pas la différence en mémoire entre instructions et données car il ne possède qu'un bus pour chercher les types d'objets, cela oblige donc le désassembleur à donner son interprétation du contenu mémoire. Notre désassembleur essaiera toujours d'interpréter la mémoire en terme de programme... Ainsi, il fera tout pour associer aux octets le code d'une instruction, ce qui ne sera pas toujours exact en tenant compte des intentions initiales du programmeur. Il faudra donc lire les listings produits par le désassembleur avec un oeil critique.

Le listing est découpé en 5 parties que l'on va étudier dès à présent, publiées en deux fois.

1/ Le programme d'essai et le programme principal.
Le programme d'essai est une petite application écrite pour l'Amiga qui utilise le désassembleur et affiche le résultat dans une fenêtre. Le programme principal est le désassembleur proprement dit. Ce dernier est compatible avec les ST et autres machines à base de 68000. Tout le programme a été écrit en PC-RELATIF et se reloge tout seul. Cela veut dire que pour l'inclure dans un programme, il suffit de le charger comme des données habituelles et d'appeler la routine de désassemblage avec les paramètres suivants :
* a0= adresse de début du désassemblage
* a1= adresse de fin de désassemblage
* d1= nombre maximum de lignes à fournir. Ceci est la deuxième condition de fin avec celle imposée par a1. On peut donc soit désassembler une zone mémoire comprise entre deux adresses en mettant $7fffffff dans d1 ou désassembler un nombre de lignes donné en mettant $7fffffff dans a1.
* d0= adresse de la routine de sortie. Cette routine va être appelée à chaque fois que le désassembleur désirera sortir un caractère. On est donc libre d'envoyer le listing produit vers un quelconque device ou plus simplement vers une zone mémoire. Cette routine ne doit modifier aucun registre et reçoit le code ASCII du caractère produit par le désassembleur dans D0 (dans l'octet de poids faible évidemment).

Il ne faudra évidemment pas déplacer le programme dans la mémoire une fois qu'il s'est auto-relogé sous peine d'assister à un magnifique plantage. En sortie, a0 pointe sur l'adresse de fin du désassemblage

qui peut être utile quand on désire désassembler une zone mémoire en plusieurs fois.

2/ Le décodage des instructions.
Notre désassembleur, dans un but de réduction de la taille du listing, prend les octets en mémoire successivement, ce qui n'est pas sans poser de problèmes (pour les MOVEMs), et décode les instructions grâce à deux tables dont l'explication suit dans 4 et 5. De plus, pour les instructions de saut relatif, il fournira un désassemblage ne correspondant pas exactement avec les normes Motorola en indiquant la distance du saut après le mnémonique : BNE.S 10 au lieu de BNE 10.S.

3/ Les conversions.
De facon à donner un résultat lisible, tous les chiffres décodés sont convertis en hexadécimal grâce aux 3 routines affectées chacune à une taille d'opérande donnée (octet, mot, mot long). Attention, aucun symbole $ n'est renvoyé car toutes les sorties sont en hexa.

4/ La table des instructions.
Cette table permet de reconnaître l'instrution contenue en mémoire. Elle a le format suivant :
* 2 octets Masque : le mot lu par le désassembleur est tout d'abord masqué avec cette valeur grâce à un ET (AND) logique de façon à ne retenir que les bits représentatifs d'une famille d'instructions.
* 2 octets Valeur : ...puis il est comparé à cette valeur. S'il y a égalité, nous sommes bien en présence de cette instruction.
* 6 octets Mnémonique: le mnémonique est contenu ici en code ASCII de façon à être envoyé directement à la sortie.
* 4 octets Pointeurs sur tables d'adressage : une fois l'instruction trouvée, il faut encore décoder les opérandes, on trouve ici un pointeur qui indique où chercher dans la table suivante les informations nécessaires à ce décodage.

5/ La table des adresses.
Cette dernière table contient toutes les informations de décodage des opérandes et a été séparée de la précédente car certains décodages sont communs à différentes familles d'instructions (exemple: subi et addi).
* 4 octets Taille : pointeur sur la routine qui décriptera la taille.
* 2 octets Paramètre : paramètre éventuel pour la routine ci-dessus.
* 4 octets Opérande 1 : pointeur sur la routine qui décodera le premier opérande s'il y en a un (sinon on met un zéro).
* 4 octets Paramètre 1 : paramètres éventuels pour la routine précédente.
* 4 octets Opérande 2 : pointeur sur la routine qui décodera le deuxième opérande s'il y en a un.
* 4 octets Paramètre 2 : paramètres éventuels pour la routine précédente.

Le listing est long et pour soulager vos yeux, vos doigts et aussi votre colonne vertébrale, vous pouvez le trouver dans la disquette Amiga auprès de la Boutique de Pressimage.

Eric Brunet

```
; Désassembleur 68000      E.Brunet 1989
;  APPLICATION D'ESSAI ET PROGRAMME PRINCIPAL
;       opt     o+

 move.l #txt,ptr            ;on pointe sur le début du buffer

 MOVE.L 4,A6
 LEA dosname,A1
 MOVEQ #0,D0
 JSR -408(A6)               ;open library DOS
 MOVE.L D0,dosbase          ;sauve sa base
 BEQ er                     ;si erreur alors fin
 MOVE.L dosbase,A6
 MOVE.L #name,D1
 MOVE.L #1005,D2
 JSR -30(A6)                ;ouvre une fenêtre RAW
 MOVE.L D0,conhandle
 BEQ er                     ;si erreur alors fin
big_loop:
 move.l zone,a0             ;adresse du désassemblage
 lea $7fffffff,a1           ;adresse de fin sans importance
 move.l #16,d1              ;mais on veut 16 lignes pile.
 move.l #sortie,d0          ;pointe sur routine de sortie
 bsr Desassemble            ;désassemble la zone
 move.l a0,zone2            ;sauve l'adresse de fin

 move.l dosbase,a6
 move.l conhandle,d1
 move.l #entree,d2
 moveq.l #1,d3
 jsr -42(a6)                ;entrée d'un caractère au clavier
```

```
 lea entree,a0
 move.b (a0)+,d0
 cmp.b #"x",d0              ;si on a tapé eXit, on quitte
 beq fini
 cmp.b #"u",d0              ;si on a tapé Up, on monte de 32 oct.
 bne non_up
 sub.l #32,zone
 bra non_down
non_up:
 cmp.b #"d",d0
 bne non_down
 move.l zone2,zone          ;si on a tapé Down, on
                            ;continue le désassemblage.
non_down:
 bra big_loop
fini:
 move.l dosbase,a6
 MOVE.L conhandle,D1
 JSR -36(A6)                ;ferme la fenêtre
 bra fin
er:
 move.l #-1,d0
fin:
 move.l d0,d1
 move.l dosbase,a6
 jsr -144(a6)               ;exit
 rts
; Routine de sortie de caractères, qui stocke les
; sorties jusqu'à tomber sur un retour chariot
sortie:
 cmp.b #10,d0               ;retour chariot ?
 bne save                   ;sinon on sauve les caractères
```

```
bsr save
MOVEM.L D0-A6,-(SP)
MOVE.L dosbase,A6
MOVE.L conhandle,D1
MOVE.B D0,txt
MOVE.L #txt,D2
move.l ptr,d3
sub.l d2,d3
subq #1,d3
move.l d2,ptr
JSR -48(A6)          ;on envoie le tampon dans notre fenêtre
MOVEM.L (SP)+,D0-A6
rts
save:
MOVEM.L D0-A6,-(SP)   ;on en sauve trop, je sais...
move.l ptr,a0
move.b d0,(a0)+       ;on sauve le caractère dans un tampon
move.l a0,ptr
MOVEM.L (SP)+,D0-A6
rts

dosname:
DC.B "dos.library",0
EVEN
name: ;texte venant en haut fenêtre RAW et taille de fenêtre
DC.B "RAW:8/8/600/144/"
dc.b "** L'Oiseau Vert 1989 E.Brunet"
dc.b " ** uP,dOWN,ExIT **",0
EVEN
zone:
dc.l $fc0000   ;on désassemble les ROMs Amiga (sacrilège !)
zone2:
ds.l 1
entree:
ds.b 2                ;entrée de 1 caractère
txt:
ds.b 256              ;tampon de sortie
even
dosbase:
DS.l 1
conhandle:
DS.l 1                ;notre fenêtre a son numéro Handle
ptr:ds.l 1

err:ds.l 2

;       ****************************************
;       *       Désassembleur MC 68000         *
;       *  L'Oiseau Vert Juillet 1989. E.Brunet *
;       ****************************************
; Ce programme fonctionne en Mode PC relatif, il se
; reloge donc tout seul et de manière transparente.

s:                    ;Base de relogement automatique

; Taille prise pour décrire une instruction
taille_instr          equ     14
; Taille prise pour décrire un mode d'adressage
taille_adressage      equ     22

RdsCount set 0
Rdsw: macro
ds.w 1
\1 equ RdsCount
RdsCount set RdsCount+2
endm
Rdsl: macro
ds.l 1
\1 equ RdsCount
RdsCount set RdsCount+4
endm
; Désassemble une zone mémoire. En entrée on a :
; a0=Debut de désassemblage, a1=Fin de désassemblage
; d0=adresse routine de sortie. Elle reçoit le code
; ascii dans d0 et ne doit modifier aucun registre
; d1=nombre de lignes maximum à désassembler
; En sortie on reçoit : a0=Adresse de fin de désassemblage

Desassemble:
bsr init_desass                 ;prépare le désassemblage
Boucle_desassemblage:
bsr desassemble_une_ligne
cmp.l adr_fin_desassemblage(a6),a0      ;Fin si
bpl fin_desassemblage                   ;adr fin dépassée
subq.l #1,nb_lignes_max(a6)             ;ou si nombre
bgt Boucle_desassemblage                ;de lignes dépassé
fin_desassemblage:
rts                                     ;retour
;
; Cette routine prépare le désassembleur
; et le reloge à l'adresse où il a été chargé.
;
init_desass:                    ;prend une base relogeable
lea var_base(pc),a6             ;Sauve les paramètres
move.l a1,adr_fin_desassemblage(a6)
move.l d0,output(a6)
move.l d1,nb_lignes_max(a6)

tst.w flag_init(a6) ;Si init déjà faite, ne pas recommencer
bne l_init_est_deja_faite

movem.l d0-a6,-(sp)

st flag_init(a6)        ;on se protège d'une nouvelle init
lea s(pc),a1            ;on prend le début du programme
move.l a1,d1

lea table_instructions(pc),a2 ;reloge la table instructions
lea fin_table_instructions(pc),a4
loop_relog_table_instr:
add.l d1,10(a2)     ;reloge le pointeur sur table adressage
add.w #taille_instr,a2
cmp.l a4,a2
bmi loop_relog_table_instr

lea table_adressages(pc),a3     ;reloge la table adressage
lea fin_table_adressages(pc),a4
loop_relog_adressages:
add.l d1,(a3)               ;reloge le pointeur sur routine
tst.l 6(a3)                 ;on évite de reloger les 0
beq ne_relog_pas_les_0_
add.l d1,6(a3)              ;reloge le pointeur sur routine
ne_relog_pas_les_0_:
tst.l 14(a3)                ;on ne reloge pas les 0
```

```
 beq ne_relog_pas_les_0__
 add.l d1,14(a3)
ne_relog_pas_les_0__:
 add.w #taille_adressage,a3
 cmp.l a4,a3
 bmi loop_relog_adressages

;on reloge aussi la table des adresses effectives
 lea table_EA+4(pc),a0
 moveq.w #14,d0
loop_relog_table_ea:
 add.l d1,(a0)                    ;on reloge l'adresse
 add.w #8,a0
 dbra d0,loop_relog_table_ea

 movem.l (sp)+,d0-a6

l_init_est_deja_faite:
 rts                              ;ça y est, tout est prêt
;
; Cette routine désassemble une instruction
; a0=Adr de l'instruction à désassembler.
desassemble_une_ligne:
 move.l a0,d7                     ;Sort l'adresse
 bsr convert_hexa32
 move.l output(a6),a5
 move.b #" ",d0
 jsr (a5)                         ;puis un espace

 move.l sp,SauveSP(a6) ;si erreur, on cherche l'instruction
 move.w (a0)+,d0       ;Récupère l'opcode
 move.w d0,d1
 lea table_instructions(pc),a2
 lea fin_table_instructions(pc),a3
Boucle_trouve_opcode:             ;Cherche l'opcode
 and.w (a2)+,d0                   ;masquage du code
 cmp.w (a2)+,d0                   ;puis comparaison
 beq opcode_trouve                ;si c'est bon, on traite
 move.w d1,d0                     ;sinon on recommence
 add.w #taille_instr-4,a2         ;instruction suivante
 cmp.l a3,a2                      ;si c'est pas la dernière
 bmi Boucle_trouve_opcode         ;alors continue

fin_desass:                  ; si l'opcode n'a pas été trouvé
 lea interrogation(pc),a5
 moveq #1,d7                 ;on sort un ?? (2 caractères)
 bsr sort_texte_sans_espace
 bsr sort_opcode             ;puis on sort le code hexa
 bra fin_normale_de_ligne    ;et un retour chariot
interrogation:
 dc.b "??"
 even

opcode_trouve:               ;on a l'opcode dans d1
 clr.b taille_presente(a6)   ;au cas où...
 move.l a2,a5
 moveq.l #5,d7               ;on sort le mnémonique
 bsr sort_texte_sans_espace ;qui fait 6 lettres maximum
 addq.l #6,a2                ;on pointe après le nom

 move.l (a2),a2  ;on va dans la table générale de adressages
 move.l (a2)+,a5             ;on cherche la taille .b/.w/.l
```

```
 move.w d1,d0                ;on appelle la routine qui
 jsr (a5)                    ;cherche ma taille

 move.l output(a6),a5        ;un espace entre mnémonique
 move.b #" ",d0              ;et opérande
 jsr (a5)

 move.l (a2)+,a5             ;on cherche l'op1
 cmp.w #0,a5                 ;si pas d'opérande alors
 beq pas_d_operande2         ;on a fini
 move.w d1,d0                ;on appelle la routine qui cherche
 jsr (a5)                    ;l'opérande 1

 move.l (a2)+,a4             ;on cherche l'op2
 cmp.w #0,a4                 ;à moins qu'il n'y en ait pas besoin
 beq pas_d_operande2

 move.b #",",d0              ;met un virgule entre op1 et op2
 move.l output(a6),a5
 jsr (a5)

 move.w d1,d0                ;appelle la routine qui
 jsr (a4)                    ;cherche l'opérande 2

pas_d_operande2:
fin_normale_de_ligne:
 move.b #10,d0               ;retour chariot
 move.l output(a6),a5        ;en fin de ligne
 jsr (a5)
 rts                         ;on a désassemblé une instruction
;
;     DECODAGE DES INSTRUCTIONS
;   Routines recherchant la taille
;
pas_de_taille:               ;pas de taille à chercher
 addq.w #2,a2                ;on pointe sur la suite
 rts

taille_secrete:   ;taille n'apparaît pas dans désassemblage
 move.w (a2)+,taille_presente(a6)        ;mais on la sauve
 rts

table_taille_type_move:       ;%00=RIEN %01=.B %10=.L %11=.W
 dc.b "  ",".b",".l",".w"     ;textes possibles .b .w .l
taille_type_move:     ;traite les instructions qui décodent
 lea table_taille_type_move(pc),a5    ;la taille comme MOVE
 bra trouve_taille

table_taille_type_addi:        ;00=.B 01=.W 10=.L 11=RIEN
 dc.b ".b",".w",".l","  "
taille_type_addi:     ;traite les instructions qui décodent
 lea table_taille_type_addi(pc),a5    ;la taille comme ADDI
 bra trouve_taille

trouve_taille:    ;routine générale récupération de taille
 move.w (a2)+,d6  ;récupère le No des bits contenant taille
 lsr.w d6,d0                 ;dans l'opcode puis on récupère
 and.w #%11,d0               ;la taille dans D0*2
 add.w d0,d0                 ;pour avoir une adresse paire
trouve_taille_GENERAL:
 add.w d0,a5                 ;a5 pointe sur le texte .b .w ou .l
 cmp.b #" ",(a5)             ;si pas de taille alors erreur
```

```
 beq erreur
 moveq #1,d7                    ;sinon on sort le texte
 move.b 1(a5),taille_presente(a6)
 bra sort_texte_sans_espace

table_taille_type_movem:        ;table des textes pour MOVEM
 dc.b ".w",".l"                 ;0=.w 1=.l
taille_type_movem_inverse:      ;1=.w 0=.l
 not.w d0 ;des instructions utilisent cette autre convention
taille_type_movem: ;0=.w 1=.l
 lea table_taille_type_movem(pc),a5  ;taille MOVEM sur 1 bit
trouve_taille_1_bit:  ;routine globale pour instructions qui
 move.w (a2)+,d6       ;codent leur taille sur 1 bit
 lsr.w d6,d0           ;on récupère là aussi la position
 and.w #1,d0           ;du bit que l'on isole dans d0
 add.w d0,d0           ;et on trouve l'offset pair
 bra trouve_taille_GENERAL   ;on réutilise routine ci-dessus
erreur:
 move.l a0,err                  ;détectée
 move.l a2,err+4
 move.l SauveSP(a6),SP          ;arg c'est immonde mais
 bra fin_desass                 ;la cause est désespérée
;
; Routines recherchant une condition pour Bcc et Scc
;
condition:
 move.w (a2)+,d7          ;recherche des N° de bits contenant
 lsr.w d7,d0              ;la condition que l'on isole
 and.w #%1111,d0
 add.w d0,d0              ;et qui nous servent pour calculer
 lea table_cond(pc,d0.w),a5          ;l'adresse du texte
 moveq #1,d7              ;à 2 caractères qu'il faut sortir
 bra sort_texte_sans_espace
table_cond:               ;table des textes des conditions
 dc.b "T ","F ","HI","LS"
 dc.b "CC","CS","NE","EQ"
 dc.b "VC","VS","PL","MI"
 dc.b "GE","LT","GT","LE"
 even
;
; Routines recherchant l'opérande
; opcode inconnu type DC.W
sort_opcode:
 move.w d0,d7                        ;sort l'opcode
 bra convert_hexa16

; textes directs type move to CCR
texte_direct:
 move.l a2,a5
 addq #4,a2              ;on sort un texte de 4 lettres
 moveq #3,d7             ;contenu dans la table des adressages
 bra sort_texte_sans_espace

; Sauts relatifs Bcc
short_texte:             ;texte à sortir si branchement court
 dc.b ".s "              ;nous devions respecter ici des
                         ;conventions : Bcc.s adr = Bcc adr.s
 even
saut_relatif_variable:
 tst.b d0                ;détecte si l'adresse est longue ou non
 beq saut_relatif_long ;en testant l'octet faible de l'opcode
 move.b d0,d7            ;si c'est court alors on récupère
 ext.w d7                ;le déplacement que l'on étend à 32 bits
 ext.l d7                ;pour l'ajouter à l'adresse actuelle
                         ;de désassemblage qui est égale à celle
 add.l a0,d7             ;qu'a le ProgramCountor lors d'une
 addq #4,a2                      ;exécution du programme
 bsr convert_hexa32              ;on sort l'adresse puis
 lea short_texte(pc),a5          ;on sort le .s pour short
 bra sort_texte_sans_espace

saut_relatif_long:
 move.w (a0)+,d7          ;si l'adressage est relatif long
 lea -2(a0),a3            ;on reprend le déplacement dans d7
 ext.l d7                 ;on l'étend à 32 bits et on
 add.l a3,d7              ;ajoute l'adresse du désassemblage
 addq #4,a2
 bra convert_hexa32       ;on sort directement le texte

; Liste de registres (MOVEM)
; Cette premiere routine s'occupe des MOVEMs
; qui prennent les registres dans l'ordre D0 à A7
liste_reg_d0_a7:
 moveq.l #0,d3            ;afin de réutiliser la routine
 move.w (a0)+,d7          ;déjà faite ci-dessous. Puis on
 moveq.w #15,d2           ;va renverser les bits de l'opérande
loop_revert:
 roxr.w #1,d7             ;ces 2 instructions renversent D7
 roxl.w #1,d3             ;et mettent le résultat dans d3
 dbra d2,loop_revert
 bra affiche_liste        ;affiche les registres
;
; Cette deuxième routine s'occupe des MOVEMs
; qui prennent les registres de A7 à D0
liste_reg_a7_d0:
 moveq.l #0,d3            ;ici pas de renversement
 move.w (a0)+,d3          ;on récupère juste l'opérande
 bra.s affiche_liste
liste_reg_a7_d0_MOVEM_mem_reg:
 moveq.l #0,d3            ;ici pas de renversement
 move.l SauveADR(a6),a5
 move.w (a5),d3
; Cette routine affiche une liste de registres
affiche_liste:
 lea table_index(pc),a5   ;on réutilise une table
                ;qui contient déjà tous les noms de registres
 moveq #15,d2             ;16 tours pour 16 registres
 moveq #1,d7
 move.l output(a6),a4
 clr.b d4                 ;flag mettre slash = non qui
          ;évite de mettre un / avant le 1er registre cité
loop_affiche_liste:
 ror.w #1,d3              ;registre demandé ?
 bcc pas_de_ce_reg        ;sinon alors rien
 tst.b d4                 ;sinon teste s'il faut un /
 beq pas_de_slash         ;si pas de / alors saute
 move.b #"/",d0           ;sinon sort le /
 jsr (a4)
pas_de_slash:
 bsr sort_texte_sans_espace
 st d4 ;maintenant il en faudra un à chaque nouveau registre
pas_de_ce_reg:            ;registre non demandé
 add.w #8,a5              ;saute 8 octets dans la table des noms
 dbra d2,loop_affiche_liste                ;et continue
```

```
 addq.w #4,a2
 rts
;
; Immédiat #. Voir mode_IMM
; Donnée placée dans l'opcode (1 à 8 bits)
; Cela sert pour les MOVEQ, ADDQ...
little_data:
 move.b #"#",d0              ;sort le caractère #
 move.l output(a6),a5
 jsr (a5)
 move.l d1,d0                ;récupère l'opcode et
 move.w (a2)+,d7             ;un paramètre de désassemblage
 bmi ChiffreNulEquHuit       ;si <0 alors saute, sinon
 lsr.w d7,d0                 ;on ne fait qu'isoler les bits
 and.w (a2)+,d0              ;de l'opérande
ChiffreNonNul:
 move.w d0,d7                ;on sort l'opérande
 bra convert_hexa8
;
; Certaines instructions, dont l'opérande est sur 4
; bits, considèrent le 0 comme un 8 : MOVEQ #8 se
; code comme si on avait MOVEQ #0
ChiffreNulEquHuit:
 neg.w d7                    ;on rend positif le paramètre
 lsr.w d7,d0                 ;on isole les bits de l'opcode
 and.w (a2)+,d0              ;et on les masque
 bne ChiffreNonNul           ;si c'est nul
 move.w #8,d7                ;alors on sort directement 8
 bra convert_hexa8

; Registre de donnée Dn
reg_data:
 move.w (a2)+,d7             ;on récupère les numéros de bits
 addq.w #2,a2                ;contenant le numéro de registre
 lsr.w d7,d0                 ;puis on l'isole grâce au AND
 and.w #%111,d0
 bsr modeDN                     ;on prépare le texte
 bra sort_texte_sans_espace     ;et on le sort

; Registre d'adresse An
reg_adr:
 move.w (a2)+,d7             ;on fait comme pour les
 addq.w #2,a2                ;registres de données
 lsr.w d7,d0
 and.w #%111,d0
 bsr modeAN                     ;on prépare le texte
 bra sort_texte_sans_espace     ;et on le sort

; Predécrémentation -(An)
predecrem:
 move.w (a2)+,d7 ;idem
 addq.w #2,a2
 lsr.w d7,d0
 and.w #%111,d0
 bsr mode_moins_AN_
 bra sort_texte_sans_espace
; rts

; Postincrémentation (An)+
postincrem:
 move.w (a2)+,d7 ; idem
 addq.w #2,a2
 lsr.w d7,d0
 and.w #%111,d0
 bsr mode_AN_plus
 bra sort_texte_sans_espace
; rts

; deplacement d(An)
deplacement:
 move.w (a2)+,d7 ;idem
 addq.w #2,a2
 lsr.w d7,d0
 and.w #%111,d0
 bsr mode_d_AN_
 bra sort_texte_sans_espace
; rts

; Effective adress reverse (REGISTRE,MODE). Certaines
; "effectives adresses" (EA) sont codées dans ce sens
Reff_adr:
 move.w (a2)+,d7                ;on récupère le No de bit
 lsr.w d7,d0                    ;de l'EA et on l'isole
 move.w d0,d2
 and.w #%111,d2                 ;d2=MODE=BITS 2,1,0
 and.w #%111000,d0              ;d0=REGISTRE=BITS 5,4,3
 lsr.w #3,d0
 bra on_a_le_bon_ordre

eff_adr_MOVEM_mem_reg:
 move.l a0,SauveADR(a6)
 addq.w #2,a0                ;saute 2 octets pour le movem

; Effective adress (REGISTRE,MODE)
eff_adr:
 move.w (a2)+,d7             ;pareil mais dans l'autre sens
 lsr.w d7,d0
 move.w d0,d2
 and.w #%111,d0                 ;d0=REGISTRE=BITS 2,1,0
 and.w #%111000,d2              ;D2=MODE=BITS 5,4,3
 lsr.w #3,d2
; Cette routine ne traite que le cas D0=REG,d2=MODE
on_a_le_bon_ordre:
 move.w d2,d3
 cmp.w #%111,d2      ;le mode %111 a un traitement spécial
 bne pas_mode111     ;alors on crée un mode inexistant %1000
 moveq.w #%1000,d3   ;et on ajoute les bits REGISTRE à ce mode
 or.w d0,d3
pas_mode111:   ;quand on arrive ici, le mode est codé sur 4
 lsl.w #3,d3   ;bits au lieu de 3 pour gérer le mode %111 comme
 lea table_EA(pc,d3.w),a1   ;les autres. On se sert du mode
 move.l (a1)+,d4        ;comme pointeur sur la table table_EA
 and.w (a2)+,d4         ;qui indiquera la routine à exécuter
 beq erreur    ;à moins que le mode ne soit inconnu comme
   ;le montre le test fait avec le and ci-dessus qui compare
   ;le mode avec ceux possibles pour une instruction donnée
 move.l (a1)+,a1            ;on recupère l'adresse de la
 jsr (a1)                   ;routine à exécuter et on y saute

 bra sort_texte_sans_espace ;on affiche le texte créé

; Cette table a le format suivant:
; - chiffre donnant le bit correspondant au mode (c'est 1 masque)
; - adresse de la routine traitant cette adresse effective
```

```
table_EA:
 dc.l 1,modeDN-s,2,modeAN-s,4,mode_AN_-s
 dc.l 8,mode_AN_plus-s,16,mode_moins_AN_-s
 dc.l 32,mode_d_AN_-s,64,mode_d_AN_xi_-s
 dc.l 0,0
 dc.l 128,mode_ABSw-s,256,mode_ABSl-s
 dc.l 512,mode_d_PC_-s,1024,mode_d_PC_xi_-s
 dc.l 2048,mode_IMM-s,0,0,0,0,0,0,0,0
;
; Traitement des adresses effectives. Elles renvoient la
; taille (d7) et l'adresse du texte (a5) pour permettre
; l'appel de la routine sort_texte_sans_espace
;
; mode Dn pour les MOVE Dx,Dy
modeDN:
 lea texte_mode_DN(pc),a5
 add.b #"0",d0          ;on ajoute 65 pour se mettre en ASCII
 move.b d0,1(a5)        ;on met le chiffre directement dans
 moveq #1,d7            ;le texte ci-dessous
 rts
texte_mode_DN:
 dc.b "Dn"              ;ici le n sera remplacé par un chiffre
 even

; mode An pour les MOVE Ax,Ay
modeAN:
 lea texte_mode_AN+2(pc),a5
 add.b #"0",d0          ;on fait le même traitement que pour
 move.b d0,1(a5)        ;le mode Dn mais on se décale par
 moveq #1,d7            ;rapport au début du texte pour
 rts                    ;n'afficher que An

texte_mode_AN:          ;ce texte est utilisé par 4 routines
                        ;alors il tient compte des 4 possibilités
                        ;An ou (An) ou (An)+ ou -(An) (et même d(An))
 dc.b "-(An)+"
 even

mode_AN_:               ; mode (An) pour les MOVE (Ax),(Ay)
 lea texte_mode_AN+1(pc),a5
 add.b #"0",d0
 move.b d0,2(a5)        ;on fait de même mais on affiche
 moveq #3,d7            ;un peu plus de texte
 rts

mode_AN_plus:           ; mode (An)+ pour les MOVE (Ax)+,(Ay)+
 lea texte_mode_AN+1(pc),a5
 add.b #"0",d0
 move.b d0,2(a5)        ;ici on affiche le +
 moveq #4,d7
 rts

mode_moins_AN_:         ; mode -(An) pour les MOVE -(Ax),-(Ay)
 lea texte_mode_AN(pc),a5
 add.b #"0",d0
 move.b d0,3(a5)        ;pareil (ça devient monotone)
 moveq #4,d7
 rts

mode_d_AN_:             ; mode (An) pour les MOVE (Ax),(Ay)
 move.w (a0)+,d7        ;là, par contre, on affiche le seul décalage
 bsr convert_hexa16     ;puis on réutilise la routine
 bra mode_AN_           ;qui affiche les (An)

texte_mode_d_AN_xi_:    ;texte général des
 dc.b "(An,rn.z)"       ;modes d(An,rn.z)
 even
; mode d(An,rn.z) par exemple
; MOVE.l 0(A7,A7.l),1(A0,D0.w) (Bizarre..!)
mode_d_AN_xi_:
 lea texte_mode_d_AN_xi_(pc),a5
 add.w #"0",d0          ;on prépare le code ASCII pour An
 move.b d0,2(a5)        ;et on le met dans le texte
 move.w (a0)+,d7        ;on récupère l'opérande pour
 move.w d7,d3           ;compléter le reste. Puis on
 and.w #$ff,d7          ;cherche le déplacement sur 8
 bsr convert_hexa8      ;bits et on l'affiche
;on reçoit ici soit le cas d_An_xi_ soit le d_PC_xi_
mode_d_RN_xi_:
 lsr.w #8,d3            ;on décale de 9 bits pour récupérer
                        ;le registre Rn et avoir l'adresse du
 lsr.w #1,d3            ;texte qui complètera le texte final
 and.w #%1111100,d3     ;contenu dans texte_mode_d_AN_xi_
 move.l table_index(pc,d3.w),4(a5) ;recopie le texte de 4 oct
 moveq #8,d7            ;demande une sortie de 9 caractères
 rts
table_index:
 dc.b "D0.wD0.l","D1.wD1.l","D2.wD2.l","D3.wD3.l"
 dc.b "D4.wD4.l","D5.wD5.l","D6.wD6.l","D7.wD7.l"
 dc.b "A0.wA0.l","A1.wA1.l","A2.wA2.l","A3.wA3.l"
 dc.b "A4.wA4.l","A5.wA5.l","A6.wA6.l","A7.wA7.l"

pointW equ table_index+2        ;adresse du texte .w
; Mode absolu court ex: JMP 4.w
mode_ABSw:
 move.w (a0)+,d7
 bsr convert_hexa16             ;sort le nombre sur 16 bits
 lea pointW(pc),a5
 moveq.l #1,d7
 rts

pointL equ table_index+6        ;adresse du texte .l
; Mode absolu long ex: JMP $70000.l (il y a souvent
; des choses à cette adresse)
mode_ABSl:
 move.l (a0)+,d7
 bsr convert_hexa32             ;sort un nombre sur 32 bits
 lea pointL(pc),a5
 moveq.l #1,d7
 rts

texte_mode_PC_:
 dc.b "(PC)"
 even
mode_d_PC_:
 move.w (a0)+,d7        ;récupère le décalage et étends-le
 ext.l d7               ;à 32 bits puis ajoutes-le à l'adr.
 add.l a0,d7            ;actuelle du désassemblage puis
 subq #2,d7             ;enlèves 2 car le PC pointe sur
 bsr convert_hexa32     ;l'opérande et non sur l'opcode
 lea texte_mode_PC_(pc),a5      ;puis on sort le (PC)
 moveq #3,d7                    ;qui fait 4 caractères
 rts
```

```
texte_mode_d_PC_xi_:    ;texte général du mode d(PC,Rn)
 dc.b "(PC,rn.z)"
 even
mode_d_PC_xi_:
 move.w (a0)+,d7        ;on récupère le décalage sur 8 bits
 move.w d7,d3
 ext.w d7      ;et on l'étend sur 32 bits avant de le sortir
 ext.l d7
 add.l a0,d7
 subq #2,d7         ;on tient aussi compte de l'avance du PC
 bsr convert_hexa32
 lea texte_mode_d_PC_xi_(pc),a5
 bra mode_d_RN_xi_ ;on réutilise routine qui affiche le Rn.z
texte_vide:
 dc.b " "
 even
immediat:
 addq.l #4,a2
mode_IMM:
 move.b #"#",d0      ;on sort un # pour l'adressage immédiat
 move.l output(a6),a5
 jsr (a5)              ;puis texte vide car cet adressage n'a
 lea texte_vide(pc),a5 ;aucun texte personnel selon la taille
 move.b taille_presente(a6),d3   ;de l'instruction
 beq erreur                      ;et s'il n'y a pas d'erreur
 cmp.b #"b",d3
 beq diese_pointB           ;on appelle la routine concernée
 cmp.b #"w",d3
 beq diese_pointW
 cmp.b #"l",d3
 beq diese_pointL
 bra erreur                 ;on prend des précautions
; Ces 3 routines sortent un opérande d'une certaine taille
diese_pointB:
 move.w (a0)+,d7        ;l'opérande .b est codé comme un .w
 bra convert_hexa8

diese_pointW:
 move.w (a0)+,d7
 bra convert_hexa16

diese_pointL:
 move.l (a0)+,d7
 bra convert_hexa32
;
;       LES ROUTINES DE SORTIE
; Les 3 routines suivantes sortent de l'hexadécimal
; d7=nombre à sortir sur 32 bits
; a6=adresse de base des variables
convert_hexa32:
 movem.l d0-d7/a0-a6,-(sp)     ;on sauve tous les registres
 moveq #7,d2                   ;8 tours pour 8 quartets
 lea table_convert(pc),a1      ;on pointe sur notre table
 move.l output(a6),a5  ;prépare le saut à la routine de sortie
boucle_convert32:
 rol.l #4,d7            ;prend le quartet (ou nibble) suivant
 move.l d7,d1           ;on l'isole dans d1
 and.l #15,d1
 move.b 0(a1,d1.w),d0          ;et on cherche son code ASCII
 jsr (a5)                      ;avant de faire la sortie
 dbra d2,boucle_convert32
 movem.l (sp)+,d0-d7/a0-a6     ;ni vu ni connu...
 rts
; d7=nombre à sortir sur 16 bits
; a6=adresse de base des variables
convert_hexa16:
 movem.l d0-d7/a0-a6,-(sp)
 moveq #3,d2                        ;4 quartets à sortir
 lea table_convert(pc),a1
 move.l output(a6),a5
boucle_convert16:
 rol.w #4,d7                  ;on fait la rotation sur 16 bits
 move.l d7,d1
 and.l #15,d1
 move.b 0(a1,d1.w),d0
 jsr (a5)
 dbra d2,boucle_convert16
 movem.l (sp)+,d0-d7/a0-a6
 rts
; d7=nombre à sortir sur 8 bits
; a6=adresse de base des variables
convert_hexa8:
 movem.l d0-d7/a0-a6,-(sp)
 moveq #1,d2 ;2 quartets à sortir
 lea table_convert(pc),a1
 move.l output(a6),a5
boucle_convert8:
 rol.b #4,d7
 move.l d7,d1
 and.l #15,d1
 move.b 0(a1,d1.w),d0
 jsr (a5)                           ;sortie du carac
 dbra d2,boucle_convert8
 movem.l (sp)+,d0-d7/a0-a6
 rts
table_convert:
 dc.b "0123456789ABCDEF"

; Sort un texte d'une longueur donnée mais s'arrête si elle
; rencontre un espace. a5=adr texte,d7=longueur max-1
sort_texte_sans_espace:
 movem.l d0-d7/a0-a6,-(sp)      ;là encore, on sauve tout
 move.l output(a6),a4           ;on prépare la sortie
loop_sort_t_s_e:
 move.b (a5)+,d0          ;on prend le caractère suivant que
 cmp.b #" ",d0            ;l'on compare à un espace pour
 beq fin_sort_t_s_e       ;voir si l'on sort de la boucle
 jsr (a4)                 ;on envoie le caractère
 dbra d7,loop_sort_t_s_e        ;on tourne D7+1 fois
fin_sort_t_s_e:
 movem.l (sp)+,d0-d7/a0-a6      ;et on récupère tout
 rts
; Variables système. Ces variables sont stockées de manière
; relative à var_base, aussi leurs labels ne sont que leur
; offset par rapport à var_base
var_base:
 Rdsl adr_fin_desassemblage     ;adr où s'arrêter
 Rdsl output                    ;adr routine de sortie
 Rdsl nb_lignes_max             ;2e condition d'arrêt
 Rdsw flag_init           ;empêche 2 relogements successifs
 Rdsw taille_presente           ;taille de l'instruction
 Rdsl SauveSP                   ;utilise en cas d'erreur
 Rdsl SauveADR                  ;utilise pour le movem
                          ; suite et fin le mois prochain...
```

NEWS

LE GST 40 A

La société Satellite et Télévision (SATV) doit commercialiser très prochainement un nouveau type de genlock incrustateur. Il se distingue des autres par son rapport qualité/prix exceptionnel. Son prix sera en effet inférieur à 2300 francs, le rendant abordable par une plus grande famille d'utilisateurs.
Ses performances, dues à un concept intelligent, peuvent faire pâlir de jalousie des modèles nettement plus sophistiqués. En effet, le signal vidéo à incruster n'est à aucun moment traité, ce qui garantit une image de sortie absolument identique à celle d'origine. Un seul mode d'incrustation est possible (l'Amiga étant sur l'image vidéo), mais il est d'excellente qualité. Le GST 40 possède une détection automatique de signal vidéo en entrée, qu'il peut substituer lors d'une rupture de celui-ci. C'est le cas lors des différentes manipulations d'un magnétoscope (rembobinage ou arrêt momentané), et évite ainsi tous plantages du système. Cette technique permet par ailleurs de remplacer de façon économique un encodeur PAL, il suffit simplement de ne pas mettre de vidéo en entrée, et vous pourrez transférer directement sur magnétoscope (PAL uniquement) vos animations ou dessins.
Une simplicité de mise en oeuvre hors du commun font du GST 40 Amiga un outil idéal pour ceux dont l'exigence de qualité est tributaire d'un budget serré ou des connaissances techniques moindres. Une version uniquement SuperVHS devrait, elle aussi, voir le jour.

PERFORMER 2.0

Une seconde version plus complète de ce logiciel de présentation graphique, que nous avions visité avec vous le mois dernier, est annoncée pour la rentrée. Elle devrait convaincre les professionnels de l'image, et offrir grâce à ses nouvelles ouvertures un champ d'applications encore plus vaste. La "2.0" accepte encore plus de formats graphiques, notamment les fichiers RGB 8, RGB N, RGB 24, et surtout le "new movie" utilisé par Sculpt. Pour ce dernier, il permet entre autres de le décompacter et d'en extraire les images.
Cette version est aussi compatible avec le Workbench 2.0 et peut être pilotée soit par ARexx, soit par le MIDI. D'autre part, un module spécial de pilotage sera intégré au séquenceur musical Bars & Pipes, voilà de quoi séduire les mordus du multimédia. L'importateur français a trouvé un palliatif à un bug de l'Amiga, qui interdisait l'affichage des images haute résolution Pal overscan au-delà de 8 couleurs. Il vous suffit de venir dans les "préférences" et d'opérer un recentrage de l'écran (vers la gauche), et de sauvegarder lesdites préférences. Et zoup, le tour est joué ! Cette astuce est aussi valable avec les autres logiciels ayant ce type de problème, merci CIS. Pour revenir à cette nouvelle version de Performer, une application ARexx est offerte avec, et permet d'enregistrer en temps réel une séquence d'images ou d'animation et de les "rejouer" avec un timing exact. Vous pourrez acquérir ce superbe outil pour 950 F environ.

MMU 1.0 et PERFECT SOUND III

MMU (Music Management Unit) est un logiciel d'assemblage et d'édition d'échantillons sonores très complet. La version 1.0 de ce soft sera offerte gratuitement avec Perfect Sound III. Un interfaçage direct entre ces deux produits offre une puissance et une palette de possibilités étonnante. Vous pourrez acquérir très prochainement cet ensemble pour un peu moins de 800 F, toujours distribué par CIS.

ÇA VAUT LE COUP !

Le même importateur nous communique en dernière minute une offre commerciale promotionnelle imposible à passer sous silence. L'acheteur du fameux Kit pour Amiga 2000 : A 3001, équipé d'un 68030 à 28MHz, d'un coprocesseur arithmétique 68882, de 4Mo de mémoire 32 bits et d'une interface pour disque dur, se verra offrir le disque dur, et pas n'importe lequel puisqu'il s'agit d'un Quantum 40Mo ! Cette offre, valable pendant tout l'été, permet de réaliser une économie substancielle de 6000 F environ, soit le tout pour un tout petit peu moins de 20000 F.

DESIGN...

La société Bus+ propose une carrosserie "tower" destinée à recevoir un Amiga 2000 ainsi que différents périphériques. De construction allemande traditionnelle, rien n'a été laissé au hasard. L'alimentation a été décalée vers le haut, permettant ainsi de disposer d'une place non négligeable entre les cartes et elle-même. Les faces avant et arrière sont en plastique moulé. On remarque tout de suite que les interrupteurs de façade ne correspondent à rien de concret sur l'Amiga, ce dernier

n'ayant que faire du reset, ou du mode turbo. Par contre, ils peuvent être pratiques pour d'éventuelles mises en route de périphériques internes... Toute la visserie et surtout les différents connecteurs sont aussi fournis. Le prix, avoisinant les 3400 F, peut dissuader les amateurs de look, mais reste justifié pour ceux qui possèdent une configuration importante, ou ceux qui ont intégré l'Amiga dans leurs tâches quotidiennes.

Profitons-en aussi pour annoncer la disponibilité, chez le même importateur, d'un boîtier destiné à l'Amiga 500. Le "Checkmate A1500" peut recevoir un Amiga 500, et possède les emplacements pour un disque dur A590, et deux lecteurs internes. Le joystick, la souris et le clavier dorénavant détachables, se connectent en face avant. Dans le boîtier, restent les emplacements pour différents modules, encore en dévelopement, un slot vidéo pour recevoir un antifliker ou un genlock, ainsi qu'une carte "fond de panier" pour recevoir des cartes d'extension de 2000. Il vous faudra débourser environ 2900 F pour avoir un look intermédiaire entre un 1000 et un 3000...
Pour en finir avec Bus+, et pour laisser une idée d'achat à ceux qui sont moins fortunés, ils pourront, pour 450 F, acquérir un câble antiflicker fonctionnant avec un simple moniteur monochrome PC. Oh, j'allais oublier un ventilateur silencieux pour Amiga 2000, et ce pour 700 F environ.

Fabuleux !
Nous reprenons votre Amiga 2000 pour tout équipement de l'Amiga 3000

RUN INFORMATIQUE
Le plus ancien spécialiste de l'Amiga à Paris
et Commodore France
ont le plaisir de vous présenter en Avant-première leur dernière production :

AGREE A 3000

L'ordinateur qui par ses capacités graphiques et sonores va révolutionner le marché de l'informatique associée à la vidéo !

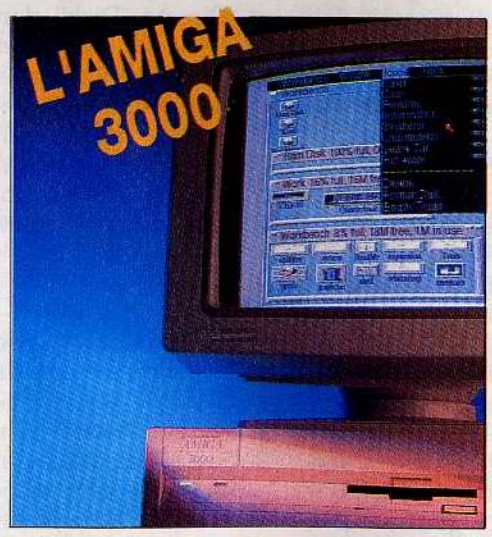
Venez découvrir ce fabuleux outil de communication qu'est l'Amiga 3000 autour de 2 Régies Vidéo dotées des plus récentes innovations technologiques Sony.

Régies animées par Thierry, pionnier de l'Amiga et Patrick de la société Point de Vue

AMIGA Vision

le logiciel multimédia le plus fabuleux du marché !
Le nec en matière de logiciel «Show, démo-tournante, ...» pour vos présentations

Appelez-nous :
Tél. : (1) 45.81.51.44
Fax : 45.88.33.45



Une production qui vous fera... ...du cinéma !!!

Services
Titrage ;
montage ;
transferts vidéo...

Entreprises
Régies Vidéo clés en main ;
Formation assurée ;
assistance permanente ;
location de
Régie Vidéo.

BON A RETOURNER A
RUN INFORMATIQUE 62, rue Gérard - 75013 Paris
Tél. : 45.81.51.44 - Fax : 45.88.33.45
DOCUMENTATION
Je désire recevoir sans engagement de ma part une documentation complète :
☐ sur l'AMIGA 3000 ☐ sur vos prestations vidéo
Société : .....
NOM : .....
Adresse : .....
Tél : .....
Fax : .....

LA FIN D'UNE POLEMIQUE ET LE DEBUT D'UNE AUTRE

En attendant la sortie fort improbable d'un émulateur Amiga sur ST, CICI annonce la disponibilité, pour début septembre, d'un émulateur ST sur notre bon Amiga. Il possède une partie Hard se glissant dans un 2000, et des versions pour 500 et 1000 seront aussi disponibles. Pour l'instant, une version tourne chez l'importateur, qui a réussi à faire fonctionner de nombreux logiciels sous toutes les résolutions, tels que Calamus, Signum 2, Arabesque, Stad, Mega Paint, le GfA basic...
Medusa (tel est son nom) peut créer sur certains disques durs une partition ST. Cet émulateur est fourni sans TOS mais avec un petit logiciel qui permet d'aller le "prendre" dans un ST, aïe... ! Le constructeur travaille actuellement sur l'interfaçage de la partie MIDI, ce qui permettrait d'avoir d'un seul coup une logithèque musicale très impressionnante (et multitâche, nananère!), mais là restons prudent...

MIAMI VICE ET VERSA

Non, il n'y a pas que des gendarmes et des voleurs qui courent partout dans les rues de Miami. Il y a aussi Hologramophone Research, qui nous propose deux softs musicaux très originaux : Hyperchord et Pixound.
Tous deux utilisent le générateur sonore de l'Amiga ainsi que le MIDI. Ils sont compatibles avec Dr.T's KCS et E.A's DMCS, pour l'édition et pour la création de partitions. Ils fonctionnent sur tout type d'Amiga disposant d'un minimum de 512K.

HYPERCHORD

Hyperchord est un séquenceur basé sur une vue atomique (les notes) et moléculaire (les motifs) de la musique, les thèmes variant en rythme, en hauteur et en ordre d'apparition. On peut ainsi créer des séquences comprenant jusqu'à 40 notes, qui seront manipulées en temps réel. Hyperchord est constitué de 4 programmes complémentaires. Mode Maker permet l'optimisation des modes tonals d'Hyperchord. Les échelles, les modes et les arpèges peuvent être affichés sous forme de "pipe" musical, ou comme des barrettes sur une corde, ou encore en cercles concentriques. De nombreux outils sont disponibles pour faciliter la génération des échelles. Rhythm Maker permet, comme son nom l'indique, de créer jusqu'à 39 rythmes, en utilisant des notes variant de la 64ème de noire à la ronde. Holistic Window est une interface graphique. Enfin, on gère le tout dans Hyperchord, où les séquences peuvent être renversées, bouclées, mixées, etc. En playback, la tonalité est à tout moment modifiable, 60 modes sont à votre disposition, du pentatonique à la 7ème mineure, en passant par les gammes mathématiques. D'autres paramètres sont également accessibles, comme le pitch, la vitesse, le rythme, le volume, l'ordre des notes, l'orchestration, l'harmonie... On peut aussi générer des séquences de façon algorithmique.

PIXOUND

Pixound est un interpréteur d'écran musical, c'est-à-dire qu'il transforme l'empreinte graphique d'une image en musique. La couleur de chaque pixel détermine un accord, et l'on bénéficie du même nombre de modes tonals que précédemment, pour "accorder" l'écran au style de musique désiré. Les graphistes découvriront ainsi une face tout à fait inhabituelle de leur travail, et pourront tout aussi bien écouter une fractale que créer l'image d'un prélude de Bach ou d'un 45 Tours des Sex Pistols... On dispose pour ce de 5 "musical painting", de 20 preset d'harmonies sur 3 voix, de 13 algorithmes générateurs d'écran, de différents modes de lecture, de modulateurs de couleurs, etc. Pixound est aussi compatible avec les instruments au format IFF.

Ces deux produits ne sont malheureusement pas encore disponibles en France, mais peuvent être commandés aux USA en contactant directement Hologramophone Research au 6225 S.W 145 Street, Miami, Florida 33158, ou en téléphonant au 19(1)305.252.2661 (Parents, surveillez vos notes de téléphone...). Prix communiqués : Hyperchord 159$, Pixound 99$.

XANADUUUUU...

Cette société spécialisée dans le son professionel, proposera en fin d'année une carte fonctionnant sur A2000 et A3000 permettant l'acquisition de données audio de très haute qualité. Cette carte d'échantillonnage fonctionnera en 16 bits sur deux canaux pouvant être utilisés indifféremment de façon indépendante, ou en stéréo. Sa vitesse d'échantillonnage est multiple : 96, 88.2, 48, 44.1, 24 et 22.05 KHz, et elle est programmable par soft. Cette possibilité autorise la conversion entre les différents standards professionnels sans perte de qualité. Des fonctionss propres aux magnétophones professionnels, tels que drop in/out, ou le rerecording entre pistes, sont déjà intégrées au hardware. Cette carte est auto-configurable et l'Amiga la gère comme une carte d'extension mémoire 128Ko. Des mots de commandes permettent, quant à eux, de déclencher les différentes fonctions disponibles, ou de paramétrer le Hard. La carte est prévue pour un fonctionnement en "Direct to Disk", dans le cas où elle serait associée avec un disque dur suffisamment rapide (couramment disponible). Elle sera fournie avec un logiciel de base simulant un magnétophone numérique professionnel et sera disponible dès le début du mois d'août pour les développeurs. Pour ceux qui ne pourraient attendre, ou qui désireraient développer une application spécifique, les spécifications hard et soft sont disponibles dès maintenant.

AMOS

L'éditeur de ce superbe langage initialement destiné à la construction de jeux demande à tous ses possesseurs de retourner le plus tôt possible leur carte de garantie, un cadeau les attend ! Une disquette pleine est en effet à leur disposition. Ils y trouveront :
- une procédure d'édition de menus ;
- Amal, qui n'est ni plus ni moins que le langage d'animation ;
- un convertisseur permettant de récupérer les fichiers Soundtracker ou Sonix ;
- le "Run-Only" pour de diffuser librement vos applications ;
- 600 sprites dessinés pour construire vos jeux.

Pour ce qui est des mois à venir, dès septembre vous pourrez gérer la 3D en formes pleines, tandis que dans le courant de l'automne, une version professionnelle de ce soft devrait voir le jour, incluant des commandes propres, entre autres, à la vidéo.

YOU KNOW WHAT ? I'M HAPPY !

Moi aussi, mon cher Droopy, de vous faire part de l'arrivée imminente des produits Disney Software en France. Ces logiciels sont le fruit d'une collaboration (très réussie!) entre Nathan Software, Titus, et Walt Disney. Chacun y a apporté le must de sa spécialité, que l'on retrouve d'ailleurs dans les trois catégories de softs proposées : le jeu, l'éducatif et les techniques d'animation.

Voici les petites merveilles que l'on trouve dans la partie "Entertainment" :

"**DICK TRACY**" est une série de 3 jeux d'arcade/action. Issu du film du même nom, où figurent entre autres Warren Beatty et Madonna, il retrace les aventures du détective. Avec ses potes Flattop et Pruneface, plongez dans les ruelles du crime, affrontez les truands de la pègre et sauvez la belle Tess.
"**DUCKTALES: THE QUEST FOR GOLD**", vous l'aurez deviné, est un jeu d'action/aventure où se succèdent des contrées exotiques pleines de pièges et de créatures aussi sombres qu'une nuit d'orage ou que la plus noire des grottes. Glomgold ou le trésor se cacheraient-il à Whatsamatterhorn?
"**ARACHNOPHOBIA**" est un action/thriller en 3D, merci Monsieur Spielberg ! Des araignées pas du tout gentilles, au venin mortel, et intelligentes de surcroît, se sont tranquillement installées en ville. Eh oui, elles ont capturé votre famille, allez donc les délivrer avant les pontes intempestives de la reine...

Pour les enfants, c'est bien sûr la grande saga des personnages de Disney :

"**MICKEY'S CROSSWORD PUZZLE MAKER**" est destiné aux 5/8 ans. L'apprentissage du vocabulaire se fait avec Mickey et ses p'tits potes, à travers la création et la reconstitution de mots croisés imbriqués d'images et de mots. L'oeuvre reconstruite toute langue dehors s'affiche ensuite sur le mur de la chambre.
"**DONALD'S ALPHABET CHASE**" est un programme pré-scolaire destiné aux 2/5 ans, axé sur la reconnaissance des lettres. L'alphabet en folie, composé de créatures féériques, sème la panique chez Donald. L'enfant doit intervenir au clavier pour faire un peu de "ménage".
"**MICKEY'S RUNAWAY ZOO**" familiarise les 2/5 ans avec les chiffres. Les animaux du zoo se sont échappés, et il faut aider Goofy et Mickey à ramener les charmantes petites bêtes chacune dans leur cage, par le biais du pavé numérique.
"**GOOFFY'S RAILAWAY EXPRESS**" stimule les 2/5 ans dans la reconnaissance des couleurs et des formes. C'est un voyage enchanteur à bord d'un train à la fumée magique, où les formes, sitôt attrappées, se transforment en animaux, plantes ou objets quotidiens.

Dans la catégorie "créativité", Disney Software a aussi frappé très fort: "**THE ANIMATION STUDIO**" est un programme de dessin et d'animation basé sur les célèbres techniques d'animations des studios Disney. La création de séquences s'effectue par transparence des cels, dont on détermine l'ordre d'apparition et le temps d'affichage. Vous pourrez intégrer vos animations dans différents fonds, et y rajouter des couleurs, de la musique ou des effets sonores. Le soft est livré avec 3 disquettes foisonnant d'animations, de musique, d'effets, et d'outils de créations.
"**DICK TRACY CRIMESTOPPERS PRINT KIT**" est la panoplie complète du parfait détective. On y trouve les personnages du film avec le héros, les bons et les méchants, tout le monde y est. C'est le logiciel idéal pour imprimer des cartes, des posters, du papier à entête, des badges, etc.

La disponibilité commerciale de ces logiciels s'échelonnera sur une période de six mois. Dès septembre, vous pourrez acquérir Animation Studio et toute la série des éducatifs. En octobre, c'est au tour des Dick Tracy, suivit de Duck Tales en novembre. Il faudra attendre jusqu'au mois de février 90 pour profiter enfinn d'Arachnophobia. Patience!

PROMOTION DU MOIS

Mr Georges FORNAY, directeur financier et administratif de Commodore France, vient d'être promu Directeur Général. Commodore France retrouve enfin sa tête après quelques mois d'absence... Mais le communiqué de presse fait aussi état d'un certain nombre de résultats, notamment financiers, de la firme française, et si l'on en croit les chiffres annoncés, c'est plutôt encourageant : 80% de croissance moyenne pour un chiffre d'affaires de près de 300 MF sur l'exercice 90, ce dernier étant compris comme la période allant de Juillet 89 à Juin 90. La seule progression du dernier trimestre (échu au 31 Mars 90) a été de 97%, soit pratiquement la "culbute" par rapport au dernier trimestre de l'exercice précédent. Qui a dit que Commodore allait mal ?

L'AMIGA 3000

Après nos premières descriptions de cette nouvelle machine, il était indispensable de vous fournir les prix des configurations désormais disponibles chez Commodore (tous ces prix sont ici annoncés en Hors Taxes). Sur la base d'un 3000 équipé de 1 méga de mémoire Fast et de 1 méga de mémoire Chip, on trouve les cas de figure suivants :
- 16 MHz, avec disque dur Quantum 40Mo/19ms : 21.990 F sans moniteur, et 25.990 F avec moniteur 14 pouces ;
- identique mais en 25 MHz : 25.990 F sans moniteur, et 29.990 F avec ;
- 25 MHz avec disque dur de 100Mo aux performances identiques : 30.990 sans moniteur et 34.990 avec ;
Vite, ma tirelire !...

UN PONT ENTRE DEUX MONDES

Cet article n'est que le début d'une série traitant de la carte PC. En effet, les utilisateurs que vous êtes ne possèdent pas toujours toutes les informations nécessaires pour tirer le meilleur parti de cette extension tant sur le plan logiciel que matériel. Nous allons aujourd'hui nous attacher à la description de la partie logicielle qui régit les échanges entre les deux mondes que sont AmigaDos et Ms-Dos.

Préambule : pour les informations traitant du mapping mémoire au niveau de la zone Amiga dédiée à la carte passerelle ainsi que des 128k de RAM double accès implantés sur cette dernière, reportez-vous à la documentation qui vous a été fournie par Commodore.

DESCRIPTION ET CONTENU DE LA LIBRAIRIE JANUS

Les différentes routines qui vont être décrites, permettent un accès de bas niveau au "system Janus" et régissent ainsi les échanges entre l'Amiga et le PC.

LES ROUTINES PUBLIQUES

* AllocJanusMem
* ChecjJanusInt
* CleanupJanusSig
* FreeJanusMem
* GetJanusStart
* GetParamOffset
* JBCopy
* JanusLock
* JanusMemBase
* JanusMemToOffset
* JanusMemType
* JanusUnLock
* SendJanusInt
* SetJanusEnable
* SetJanusHandler
* SetJanusRequest
* SetParamOffset
* SetupJanusSig

Ces routines sont contenues dans une librairie libellée Janus.library, qui est chargée par le système durant la procédure d'autoconfig. Pour l'interfaçage avec le langage C, il existe une librarie linkable "jlib.lib". Toutes les routines qui retournent une valeur, la mettront dans D0.

**** oldHandler = SetJanusHandler (jintnum, intserver)/(D0,A1)***

Cette fonction met en place un gestionnaire d'interruptions pour une une interruption Janus particulière. L'ancienne interruption est retournée. Une valeur nulle signifie qu'il n'existe pas de gestionnaire d'interruption et que les interruptions ne peuvent être traitées.

**** oldEnable = SetJanusEnable (jintnum, newvalue)/(D0,D1)***

Chaque jintnum (ou numéro d'interruption JANUS) peut être individuellement activé ou désactivé (en plus de la possibilité de désactiver le gestionnaire d'interruption). Si une interruption est désactivée, les demandes ou événements la concernant ne génèreront pas d'interruption. Ces demandes peuvent être faites par la routine SetJanusRequest. Si la valeur de NewValue est nulle, alors l'interruption est désactivée, et si la valeur est égale à 1 alors c'est qu'elle est activée, toutes les autres valeurs étant réservées. Cette routine génère une interruption si une interruption active a une demande en attente.

**** oldRequest = SetJanusRequest (jintnum, newvalue)/(D0,D1)***

Cette fonction active ou désactive une demande d'interruption de jintnum. Si newvalue est égale à 0, alors la demande est désactivée. Si newvalue est égale à 1 alors la demande est valide. Dans les autres cas, l'ancienne valeur de la demande est retournée.

**** SendJanusint(jintnum)/(D0)***

Cette routine est pratique pour les requêtes "system", c'est-à-dire que ces demandes ne sont pas directement générées par le hardware. Cet appel inscrit la demande dans la "zone d'interruption" et envoie une interruption hardware au PC.

**** CheckJanusInt(jintnum)/(D0)***

Cette routine retourne le mot d'état de la zone d'interruption. Il peut être utilisé pour savoir si le PC a déjà été avisé de l'interruption. Une valeur égale à $FF JNOINT signifie qu'aucune interruption n'est en cours, ou probablement que le PC l'a déjà traitée. Une valeur égale à $7F JSENDINT signifie que l'interruption est en attente. Toute autre valeur devra être traitée avec prudence.

**** ptr = AllocJanusMem (size, type)/(D0,D1)***

Cette fonction alloue de la mémoire dans la zone libre buffer ou paramètres, et retourne un pointeur permettant au 68000 d'adresser cette zone. Le paramètre "size" est exprimé

en octets. Si la valeur de retour est égale à 0 , cela signifie qu'il n'y a pas suffisamment de mémoire libre. Le paramètre "type" détermine quel type de zone est désiré, cela peut être soit MEM_PARAMETER ou alors MEMF_BUFFER. Vous pouvez spécifier également le type d'accès mémoire auquel fait réference le pointeur. Les quatre choix possibles sont :

- MEM_BYTEACCES
- MEM_WORDACCES
- MEM_GRAPHICACCES
- MEM_IOACCES

** FreeJanusMem (ptr, size)/(A1,D0)*

La mémoire spécifiée est libérée, quelques erreurs mineures peuvent être rencontrées, et seront signalées par le Système.

** ptr = JanusMemBase(type)/(D0)*

Cette fonction retourne l'adresse de base de la mémoire référencée par "type", se reporter à AllocJanusMem.

** type = JanusMemType(ptr)/(D0)*

Cette routine retourne le type du pointeur. Des résultats erratiques peuvent survenir si "ptr" ne pointe ni sur la mémoire buffer ni sur la mémoire paramètres.

** offset = JanusMemToOffset (ptr)/(D0)*

Si "ptr" pointe sur une zone mémoire buffer ou paramètres, l'offset du début de la mémoire à "ptr" sera retourné. C'est cette valeur qui sera utilisée par la fonction SetParamOffset( ) si c'est un bloc paramètres.

** offset = GetParamOffset (jintnum)/(D0)*

Cette fonction retourne l'offset paramètres pour l'interruption "jintnum". Par convention, une valeur égale à $FFFF signifie qu'aucun bloc paramètres n'a été déclaré.

** oldOffset = SetParamOffset (jintnum, offset)/(D0/D1)*

L'offset paramètres pour "jintnum" est codée sur les 16 bits de poids faible "d'offset". La valeur retournée est l'offset précédent. Le système n'interprète pas ce nombre, mais par convention, une valeur égale à $FFFF signifie qu'aucun bloc paramètres n'avait été déclaré précédemment.

** ptr = GetJanusStart()*

Cette routine retourne la valeur de l'adresse de base de la carte JANUS.

** setupsig = SetupJanusSig (jintnum,paramsize,paramtype)/ (D0,D1,D2,D3)*

Cette fonction couvre la plupart des besoins standard des utilisateurs. Il est probable que la majorité des utilisateurs de carte JANUS n'utilisent principalement que cette fonction ainsi que la routine CleanupJanusSig( ). Le but principal est de déclarer un gestionnaire d'interruption pour gérer nos demandes d'interruptions et transformer ceci en un signal qui sera envoyé à notre tâche. Cette manière de procéder permet d'ignorer toutes les subtilités relatives à l'écriture d'une routine d'interruption.
En premier lieu, il faut spécifier le numéro d'interruption (JINTNUM) ainsi que le numéro de signal (SIGNUM) qui nous sera renvoyé. JINTNUM sera donné via le fichier include Service.[hi], de la même manière SIGNUM sera donné via AllocSignal(-1) qui permet d'allouer un signal inutilisé. La manière de procéder pour récupérer une interruption est déterminée comme suit : la demande devra fixer une zone paramètres qui sera déclarée par sa taille (paramsize) en octets ainsi que son type (paramtype) pour pouvoir pointer sur elle.

Il y a une gestion des erreurs lorsque tout ceci est mis en oeuvre. Si la valeur de SIGNUM est égale à -1 alors la demande échoue (-1 est l'erreur retournée par AllocSignal). S'il y a déjà un gestionnaire d'interruptions alors la demande échoue. Si PARAMSIZE n'est pas égal à 0 et si une zone paramètres est déjà déclarée, alors la demande échoue. S'il est impossible d'allouer assez de mémoire alors la demande échoue aussi.
La demande retourne une valeur nulle si elle échoue, et dans le cas contraire, il y a retour d'un pointeur sur la structure SetupSig. Cette structure est définie dans Setupsig.[hi].

** CleanupJanusSig(setsupsig)/(A0)*

Cette routine annule tout ce qui a été fait par la routine SetupJanusSig.

** JanusUnlock(ptr)/(A0)*

Routine libérant un verrouillage Janus.

** JBCopy (source, destination, lenght)/(A0,A1,D0)*

Cette routine est destinée à la copie de zone mémoire.

LES FICHIERS INCLUDES

** janus.[hi]*

Il permet de s'interfacer avec la Janus.library. Les définitions présentes au sein de ce fichier sont spécifiques à l'Amiga, l'aspect le plus pratique étant les définitions d'allocation mémoire de différents types.

** janusreg.[hi]*

Il contient les constantes nécessaires à la gestion du hardware. La plupart des utilisateurs n'en ont aucun besoin.

** janusvar.[hi]*

Il contient les structures de données partagées entre l'Amiga et le PC.

** i86bloock.i*

C'est un bloc de commandes pour appeler directement les interruptions PC, et gérer le disque dur.

** services.[hi]*

Il contient les constantes relatives aux numéros d'interruptions. Ces valeurs correspondent aux paramètres JINTNUM.

** setupsig.[hi]*

Il contient les structures nécessaires aux appels à SetupJanusSig().

Voilà, j'espère que ces quelques informations auront rendu le sujet un peu moins nébuleux quant à l'interface logicielle entre ces deux mondes. Dans le prochain numéro, nous pousserons plus avant notre exploration...

INTERFACER AREXX AVEC LE GFA

Après l'interfaçage du langage C avec ARexx, passons au GfA-Basic. Aussi bizarrement que cela puisse paraître, une interface ARexx en GFA-Basic peut être réalisée plus facilement qu'en langage C. Cela se comprend si l'on admet que le C gère les chaînes de caractère plus lourdement que le GFA-Basic. Par exemple, l'avantage que possède le GFA-Basic de pouvoir tester les quatre premières lettres d'une chaîne permet la réalisation d'une structure de choix extrêmement souple.

Le fonctionnement général d'un programme interfacé ARexx est relativement simple à comprendre. La communication entre ARexx et le programme se fait par le biais d'un port de messages spécialement créé par le programme dans ce but. Une bonne convention consiste à donner comme nom à ce port celui de l'application, suivi de .rexx pour préciser qu'il s'agit d'un port destiné à ARexx. Une fois le port créé, il ne reste plus qu'à envoyer et recevoir des messages.

Pour cela, on utilise les deux fonctions GetMsg() et ReplyMsg() qui servent respectivement à recevoir des messages et à répondre aux messages reçus. Il est très important de répondre à un message car la tâche ARexx qui a envoyé le message attendra toujours l'accusé de réception avant de poursuivre et éventuellement envoyer d'autres messages. On doit par ailleurs toujours traiter le message avant d'utiliser la fonction ReplyMsg() car le message n'est notre propriété (c'est-à-dire modifiable) qu'entre le moment où on l'a reçu avec GetMsg() et celui où on l'a renvoyé avec ReplyMsg().

En général, on se contente avec ARexx d'envoyer une seule chaîne de caractères dont l'adresse se trouve à l'offset 40 de la structure RexxMsg. Celle-ci est récupérée par le GFA-Basic avec la fonction CHAR{} qui convertit une chaîne au format C en une chaîne GfA-Basic. Une fois que la chaîne est stockée dans une variable GfA-Basic, il ne reste plus qu'à étudier son contenu et agir en conséquence.

C'est de la manière dont vous aurez construit votre programme que va dépendre la suite. Les programmes les plus faciles à interfacer avec ARexx sont ceux qui sont déjà contrôlés par des menus, car il suffira de modifier légèrement les structures de choix pour que l'interfaçage ARexx fonctionne facilement. Certains affirment qu'un programme, dont les structures de contrôle sont suffisamment bien faites, peut être interfacé avec ARexx en moins de deux heures (à vos chronomètres !). Disons que ce qui risque de vous poser problème, c'est la définition des mots-clé reconnus par votre programme. Pour cela, il faut suivre quelques règles simples :

1) Ne pas utiliser de mots-clé reconnus par ARexx. Des mots-clé comme say, show ou autres risquent de vous causer des ennuis ;
2) Vos mots-clé devront être composés uniquement de lettres et à la rigueur de chiffres. Surtout évitez les espaces et les opérateurs ARexx (comme +,-, *, etc.) ;
3) Ils devront être les plus courts possible, et si possible être de même longueur. Cela vous aidera pendant la vérification. Une taille de 4 lettres serait idéale pour pouvoir tester directement les chaînes avec l'instruction CASE.

Le programme suivant vous permettra de comprendre le fonctionnement global d'un interfaçage ARexx. Il n'y manque que le traitement des erreurs qui n'est pas très compliqué à réaliser. Pour cela, il faut initialiser avec une valeur négative le champ Result1 (offset 32) avant d'effectuer ReplyMsg(). Cela est un peu ennuyeux car on est obligé de tester si le mot-clé est valide, d'envoyer ReplyMsg() puis d'appeler la routine correspondant au mot-clé. Ce programme étant très paresseux, il ne fait rien quand un mot-clé inconnu lui arrive. Le programme en C du mois précédent était plus courageux car il effectuait un véritable traitement des erreurs. C'est à vous de juger...

Comme prévu, avant de vous livrer le programme en GFA-Basic, voici un petit script ARexx qui devrait le faire fonctionner, puis vous trouverez à la suite le listing complet.

Patrick Solar

```
/* Interfaçage ARexx et GFA-Basic */
address 'GFAPort.rexx'
'rond'              /* Affiche un rond bleu, encore un ! */
'info'              /* Donne des informations */
'time'              /* Affiche l'heure */
'carre'             /* Un carré rouge */
'noir'              /* Fond de l'écran noir */
do i=1 to 10
'triangle'          /* Triangle vert */
end i
'blanc'             /* Fond de l'écran blanc */
'fin'            /* Demande au programme de s'interrompre */
```

Et maintenant le programme en GFA-Basic :

```
'
initialisations
traitement
sortie
PROCEDURE traitement
  DO
    ON MENU
    rexxmsg%=GetMsg(rexxport%)
    IF rexxmsg%>0
      arg$=CHAR{LONG{rexxmsg%+40}}
      LONG{rexxmsg%+32} = 0 ! pas d'erreur
      ~ReplyMsg(rexxmsg%)
      SELECT arg$
      CASE "time"
        time
      CASE "rond"
        rondbleu
      CASE "info"
        info
      CASE "noir"
        fondnoir
      CASE "fin"
        sortie
      DEFAULT
        IF arg$="carre"
          carrerouge
        ELSE IF arg$="triangle"
          trivert
        ELSE IF arg$="blanc"
          fondblanc
        ENDIF
      ENDSELECT
    ENDIF
  LOOP
RETURN
PROCEDURE traitemenu
  SELECT MENU(0)
  CASE 1
    time
  CASE 2
    info
  CASE 3
    sortie
  CASE 6
    rondbleu
  CASE 7
    carrerouge
  CASE 8
    trivert
  CASE 11
    fondblanc
  CASE 12
    fondnoir
  ENDSELECT
RETURN
PROCEDURE initialisations
  REM ****************** Initialisations ARexx
  ' création du port - allocation de la chaîne
  portname$="GFAPort.rexx"
  portname%=AllocMem(LEN(portname$)+1,&H10001)
  IF portname%=0
    ALERT 0,"Mémoire insuffisante",0,"Ok",a%
    sortie
  ENDIF
  CHAR{portname%}=portname$
  ' recherche si le port existe déjà
  rexxport%=FindPort(portname%)
  IF rexxport%<>0
    ALERT 0,"Le port existe déjà|Dois-je le
supprimer",2,"Ok|Non",a%
    IF a%=1
      ~DeletePort(rexxport%)
    ELSE
      sortie
    ENDIF
  ENDIF
  ' création du port de messages
  rexxport%=CreatePort(portname%,0)
  IF rexxport%=0 ! vérification
    sortie
  ENDIF
  REM ********** Initialisation écran et fenêtre
  OPENS 1,0,0,640,256,3,&H8000
  OPENW #0,0,11,640,245,&H408,&H1800
  REM ************** Initialisation des couleurs
  SETCOLOR 0,0     ! Noir
  SETCOLOR 1,&HFFF ! Blanc
  SETCOLOR 2,&HF00 ! Rouge
  SETCOLOR 3,&HF0  ! Vert
  SETCOLOR 4,&HF   ! Bleu
  SETCOLOR 5,&H888 ! Gris
  REM ***************** Initialisation des menus
  DIM entree$(20)
  i%=0
  DO
    READ entree$(i%)
    EXIT IF entree$(i%)="*"
    INC i%
  LOOP
  entree$(i%)=""
  entree$(i%+1)=""
  MENU entree$()
  MENU 13,64
  ON MENU GOSUB traitemenu
  REM ****** Initialisation des nombres aléatoires
  RANDOMIZE TIMER
RETURN
PROCEDURE sortie
  IF rexxport%<>0
```


Suite page 195

LIVRES AMIGA

Le Livre de Deluxe Paint III Editions Micro-Application

Deluxe Paint est un logiciel intéressant à plusieurs titres. On pourrait évoquer le côté historique du logiciel pionnier avec lequel l'Amiga a pu se faire un nom dans le monde du graphisme. On pourrait également parler du célèbre format IFF que ce logiciel a introduit. Intéressante également est la version PC du logiciel qui a ouvert l'échange d'images entre les PC et l'Amiga. Mais Deluxe Paint est aussi un programme qui fait partie de la panoplie du graphiste sur Amiga.

Le livre de Deluxe Paint III s'adresse à ceux qui seraient plus intéressés par l'aspect pratique de ce logiciel que par son histoire. Sa première partie donne un aperçu des fonctions de base du logiciel. Celles-ci sont expliquées de façon plus pédagogique qu'exhaustive. On peut donc affirmer sans se tromper que cette partie du livre est destinée aux débutants désireux de manier sans trop de risques le logiciel. Il est vrai que Deluxe Paint est un logiciel à "cachettes", où des fonctions sont dissimulées dans chaque recoin et que certains détails auraient pu mener les auteurs du livre dans des explications assez ardues. Quelques informations importantes manquent toutefois, par exemple le rôle du bouton droit de la souris dans certaines parties du logiciel.

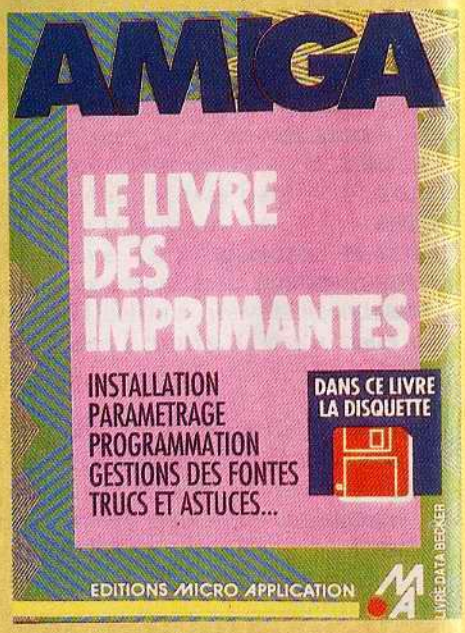
La seconde partie a reçu pour titre "Techniques et outils pour la couleur". Les aspects les plus intéressants du logiciels y sont abordés, comme les pochoirs, la perspective et l'utilisation des brosses. La partie animation occupe la troisième partie du livre et les auteurs donnent des conseils sur la manière de créer une animation avec le logiciel, en proposant également une méthode à suivre lorsque l'on veut créer une animation, à l'aide d'un exemple pratique relativement aisé à reproduire. Un chapitre contient quelques trucs et astuces, toujours intéressants à connaître et pour finir les auteurs abordent "Le point de vue de l'artiste" où sont donnés quelques conseils aux débutants en arts graphiques. Ceux-ci sont parfois étranges (comme "en mettre moins pour en retirer davantage"), mais peuvent aussi se révéler utiles. Des annexes complètent le livre en décrivant les menus et les raccourcis clavier, le programme Play et la version PC du logiciel.

Ce livre traite de manière correcte Deluxe Paint mais aurait gagné à avoir une vision un peu plus large sur la question. Deluxe Paint est un programme essentiellement destiné aux graphistes professionnels et ceux-ci n'auront que peu d'enseignements à en tirer. Il aurait été aussi bon de mentionner l'utilitaire CONVERT de la version PC qui a un grand rôle dans la conversion d'images entre PC et Amiga. Par contre, ceux que le graphisme intéresse mais qui éprouvent encore des difficultés dans l'utilisation du logiciel pourront être assistés dans leurs premiers pas grâce à ce manuel.

Le Livre des Imprimantes Editions Micro-Application

On dit souvent que les imprimantes sont les bêtes noires de l'informatique. Il est vrai que l'acquisition d'une nouvelle imprimante laisse généralement présager de 1 à 3 semaines de galères où se produiront la consommation de plusieurs kilos de papiers, le refus catégorique d'imprimer des accents, la sortie de pages blanches, des effets symptomatiques de décalage de marges, et j'en oublie. D'ailleurs, une fois que l'imprimante fonctionne, on en oublie jusqu'à son existence.

Pour s'en sortir, il vaut mieux posséder de bons logiciels de configuration (le programme Preferences suffit généralement) et une bonne documentation sur le sujet. Micro-Appli-



cation a décidé de pallier en partie le manque de documentation en ce domaine avec la publication de ce livre. Je dis en partie car il s'adresse essentiellement aux possesseurs d'imprimantes compatibles Epson, soit la grande majorité des utilisateurs d'imprimantes. L'auteur du livre, Ralf Ockenfelds, fait la démonstration d'une grande maîtrise en ce qui concerne ces imprimantes et les explications qu'il donne sont tout à fait satisfaisantes. Par contre, il aurait été souhaitable qu'il fasse preuve de la même connaissance de l'Amiga, ce qui n'est malheureusement pas le cas, et je le prouve !

Les impressions sur Amiga sont gérées au plus bas niveau par deux programmes appelés parallel.device (appelé avec PAR:) et serial.device (SER:) qui gèrent respectivement les sorties série et parallèle. En remontant un échelon, on trouve le printer.device (PRT:) qui utilise un de ces deux programmes plus un autre qui est un driver d'imprimante que l'on trouve dans le répertoire DEVS:printers. Dans Le livre des Imprimantes, l'auteur indique aux utilisateurs d'imprimantes Epson les codes de contrôles qu'ils doivent envoyer à leur imprimante pour changer sa configuration. Il explique très clairement que ceux-ci doivent utiliser les sorties SER: et PAR: pour que ces codes fonctionnent, mais sans en donner la véritable explication. Le fait est que sur Amiga les codes de contrôles ont été standardisés grâce à la sortie PRT: qui effectue une conversion d'après le driver d'imprimante sélectionné. C'est un point essentiel, car le passage en caractères gras s'effectuera avec le même code de contrôle, et cela quelle que soit l'imprimante utilisée (HP-LaserJet, Epson ou Nec). Il était donc inutile de se limiter aux imprimantes Epson et un ouvrage plus général aurait été le bienvenu. Il reste que l'auteur du livre a véritablement su détailler en profondeur le domaine auquel il s'était attaché, ce qui donne une véritable utilité à ce livre.

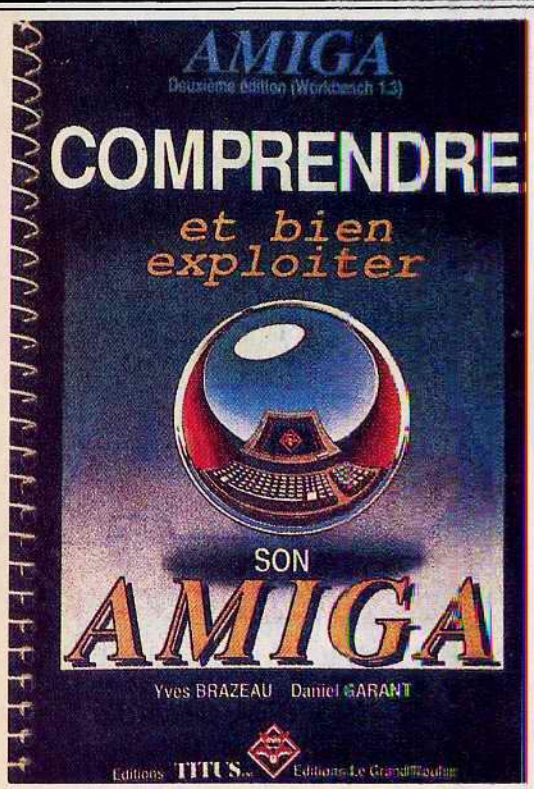
Par ailleurs, le programme Preferences est relativement bien détaillé, et on trouvera des schémas pour réaliser son câble d'imprimante, la description des fameux switches, l'explication des différents types d'imprimantes matricielles, ainsi que des solutions aux divers problèmes que peut poser une imprimante (avec parfois avec une note d'humour, exemple : "Problème numéro 10: l'imprimante couine lors de la mise en marche"...). Une disquette contient les programmes d'exemple du livre.

Pour conclure, ce livre aurait gagné à couvrir un domaine plus vaste mais apporte toutefois des informations et des solutions tout à fait dignes d'intérêt. Un programmeur ne devra pas se baser uniquement sur les informations contenues dans ce livre mais les possesseurs d'imprimantes compatibles Epson auront avantage à le consulter.

Patrick Solar

Suite de la page 193

```
    ~DeletePort(rexxport%)
  ENDIF
  IF portname%<>0
    ~FreeMem(portname%,LEN(portname$)+1)
  ENDIF
  CLOSEW #0
  CLOSES 1
  END
RETURN
PROCEDURE time
  TEXT RANDOM(500),RANDOM(150),TIME$
RETURN
PROCEDURE info
  TEXT RANDOM(500),RANDOM(150),"Menus + ARexx en GFA"
RETURN
PROCEDURE rondbleu
  COLOR 4
  PCIRCLE RANDOM(300),RANDOM(100),RANDOM(75)
RETURN
PROCEDURE carrerouge
  COLOR 2
  x=RANDOM(300)
  y=RANDOM(100)
  c=RANDOM(80)
  PBOX x,y,x+c*2,y+c
RETURN
PROCEDURE trivert
  COLOR 3
  x1=RANDOM(200)
  x2=x1+RANDOM(200)
  x3=x1+RANDOM(200)
  y1=RANDOM(100)
  y2=y1+RANDOM(50)
  y3=y2+RANDOM(50)
  LINE x1,y1,x2,y2
  LINE x2,y2,x3,y3
  LINE x1,y1,x3,y3
RETURN
PROCEDURE fondblanc
  SETCOLOR 0,&HFFF
  SETCOLOR 1,0
RETURN
PROCEDURE fondnoir
  SETCOLOR 0,0
  SETCOLOR 1,&HFFF
RETURN
DATA Projet, Heure, Information, Quitter,
DATA Objets, Rond Bleu, Carré Rouge, Triangle Vert,
DATA Fonds, Blanc, Noir, Bleu Cerise,
DATA *
```



COMPRENDRE SON AMIGA

Voici un nouveau chapitre de ce livre d'Yves BRAZEAU et Daniel GARAN édité par TITUS.

Amorce

Le Kickstart constitue l'amorce de l'ordinateur. C'est lui qui établit le noyau du système d'exploitation en mémoire (RAM) pour assurer le déroulement de l'exploitation. Sur l'Amiga 500 ou 2000, le Kickstart est interne (ROM) et invisible pour l'utilisateur. Il est en fonction dès le démarrage. Sur l'Amiga 1000, il se présente sous la forme d'une disquette qui doit être insérée dans le lecteur DF0: au moment du démarrage. Après avoir amorcé, le noyau du système d'exploitation est en fonction.

Parmi les fonctions du Kickstart nous retrouvons l'information nécessaire pour l'identification de la disquette de lancement, porteuse de la configuration de base. Si la disquette introduite après l'amorce n'est pas porteuse de cette configuration, le système d'exploitation continuera de réclamer une disquette de lancement porteuse de cette configuration. Le Kickstart au départ possède certains éléments indispensables au bon fonctionnement de la machine. Parmi les plus visibles, nous retrouvons des couleurs de base pour l'écran, un pointeur, un clavier USA0, la fonte topaz 8, etc.

Après cette phase initiale, l'ordinateur réclame l'Atelier-Workbench.

Lancement

Après avoir chargé le noyau du système d'exploitation (Kickstart), l'apparition du dessin Workbench invite l'utilisateur à introduire dans le lecteur DF0: la disquette de l'Atelier-Workbench ou toute autre disquette de lancement porteuse de la configuration de base. Le système de l'Amiga identifie la disquette de lancement comme porteuse des fichiers parfois nécessaires aux programmes utilisés subséquemment. Par conséquent, toutes les fois qu'un programme a besoin d'un fichier pour effectuer une opération, il va lire sur la disquette de lancement pour trouver ce fichier. Si la disquette de lancement est absente, une boîte de communication apparaît demandant d'introduire ladite disquette dans un des lecteurs à disquette.

Lors du lancement, les différents répertoires logiques effectifs automatiquement sont:

-le répertoire de commandes	C:
-le répertoire des sous-programmes relatifs aux périphériques	L:
-le répertoire des fichiers de commandes	S:
-le répertoire des sous-programmes divers	LIBS:
-le répertoire des périphériques	DEVS:
-le répertoire des fontes de caractères	FONTS:

Pour connaître les affectations, on utilise la commande ASSIGN en mode par commandes (voir commande ASSIGN).

Après avoir introduit la disquette de lancement, on peut modifier plusieurs des éléments prédéfinis. Le "Preferences" offre une façon de modifier, entre autres, les couleurs de l'écran et le pointeur. Lorsqu'on sauvegarde pour la première fois certaines options du "Preferences", un nouveau fichier est créé "System-Configuration" dans le sous-répertoire Devs/clipboars. Par la suite, ce fichier est mis à jour chaque fois qu'on sauvegarde à nouveau. Ce sera donc ce nouveau fichier qui prendra la relève aussitôt que cette disquette sera introduite comme disquette de lancement.

Comprendre

Le système utilise ce qui lui a été assigné au départ pour travailler. Si on demande une nouvelle fonte de caractères, si on veut imprimer, etc., le système doit rechercher l'information pour effectuer l'opération sur la disquette de lancement. Il est possible de modifier les affectations de départ en se servant de la commande ASSIGN.

Séquence de lancement

Le système Kickstart est responsable de charger le noyau du système d'exploitation en mémoire centrale afin de permettre l'exploitation de programmes de traitement. L'Atelier-Workbench est une disquette de lancement, mais toute disquette porteuse de la configuration de base peut assumer la même fonction.

Après avoir introduit la disquette Atelier-Workbench (ou une autre disquette) dans le lecteur DF0:, les directives commencent par faire succéder plusieurs messages à l'écran pour finalement s'interrompre et laisser apparaître une petite icône représentant la disquette de lancement. Il est maintenant possible d'utiliser la souris.

Que s'est-il passé entre l'insertion de la disquette et l'apparition de son icône à l'écran? Examinons le déroulement en détail.

Immédiatement après avoir introduit la disquette, le noyau du système d'exploitation vérifie s'il s'agit d'une disquette porteuse de la configuration de base (voir commande INSTALL) et ensuite le lecteur à disquette reçoit l'ordre de rechercher sur cette disquette le fichier de la séquence de lancement. Ce fichier, responsable des directives initiales, est situé dans le répertoire S et porte le nom de "Startup-sequence". A noter qu'il ne faut jamais changer le nom de ce fichier.

La séquence de lancement (Startup-sequence)

Ce fichier est un fichier de texte, constitué d'une liste séquentielle de commandes que l'ordinateur doit exécuter pour devenir fonctionnel, autant avec le mode Intuition qu'avec le mode par commandes.

Voici la liste des commandes de la séquence de lancement "Startup-sequence" de l'Atelier-Workbench 1.3. Elle-même, lors de son exécution, fait appel au fichier "Startup11" que nous listons aussi.

Le fichier de commandes "Startup-sequence"

```
Addbuffers df0: 10
c:SetPatch >NIL: ;patch system functions
cd c:
echo "A500/A2000 Workbench disk. Release 1.3 version 34.20"N"
Sys:System/FastMemFirst ;move C00000 memory to last in list
BindDrivers
SetClock load ;load system time from real time clock (A1000 owners
              ;should replace the SetClock load with Date
FF >NIL: -0 ;speed up Text
resident CLI L:Shell-Seg SYSTEM pure add ;activate Shell
resident c:Execute pure
mount newcon:
;
failat 11
run execute s:Startup11 ;This lets resident be used for rest of script
wait >NIL: 5 mins ;wait for Startup11 to complete (will signal when done)
;
SYS:System/SetMap usa1 ;Activate the ()/" on keypad
path ram: c: sys:utilities sys:system s: sys:prefs add ;set path for Workbench
LoadWB delay ;wait for inhibit to end before continuing
endcli >NIL:
```

Le fichier de commandes "Startup11"

```
resident c:Resident pure
resident c:List pure ;pre-load LIST and CD
resident c:CD pure
resident c:Mount pure ;the next 3 are loaded for speed during startup
resident c:Assign pure
resident c:Makedir pure
;make IF, ENDIF, ELSE, SKIP, ENDSKIP, and ECHO resident if
;you use scripts much, and can afford the ram.
;also make Failat, WAIT, and ENDCLI resident if you use IconX a lot
makedir ram:t
assign T: ram:t ;set up T: directory for scripts
makedir ram:env ;set up ENV: directory
assign ENV: ram:env
makedir ram:clipboards ;set up CLIPS: assign
assign CLIPS: ram:clipboards
mount speak: ;just mounting doesn't take much ram at all
mount aux:
mount pipe:
resident Mount remove ;if you have enough ram, keep these resident
resident Assign remove ;by removing these lines
resident Makedir remove
;
break 1 C ;signal to other process its ok to finish
```

Accès au fichier de la séquence de lancement

Le fichier de la séquence de lancement est un simple fichier de texte, donc accessible à tout éditeur de texte. Dans les volumes d'utilisation de l'Amiga, il est suggéré d'utiliser l'éditeur ED ou EDIT pour accéder à ce fichier et le modifier. Comme ces éditeurs sont peu pratiques et complexes, nous conseillons plutôt de se servir d'un bon programme d'édition de texte capable de sauvegarder un texte en codes ASCII.

Après avoir activé un de ces programmes, on lui indique qu'on désire le fichier "Startup-sequence" situé dans le répertoire S de la disquette Atelier-Workbench:

df0:S/Startup-sequence

Compréhension des fichiers de la séquence de lancement

"Startup-sequence"

Voici ce que signifie chacune des lignes de ce fichier de commandes:

Addbuffers df0:10

Alloue des tampons supplémentaires de mémoire au lecteur DF0:, ce qui permet d'améliorer le temps d'accès aux fichiers du disque.

C:SetPatch >NIL: ;patch system functions

Met en route la commande SETPATCH située dans le répertoire C. Cette commande a pour fonction de corriger certaines erreurs connues des Kickstart 1.2 et 1.3. Le ";" sert à inscrire, à titre d'information, la note "patch system function".

cd c:

La commande CD change le répertoire courant qui devient le répertoire C.

Echo "A500/A2000 Workbench disk. Release 1.3 version 34.20"N"

La commande ECHO donne l'ordre à l'ordinateur d'afficher à l'écran le message écrit entre guillemets.

Sys:System/FastMemFirst ;move C00000 memory to last in list

Active le programme FastMemFirst situé dans le répertoire "System". Ce programme indique au système d'utiliser d'abord la mémoire "Fast" (voir Mémoire, processeur et coprocesseur). Le ";" sert à ajouter la note "move C00000 memory to last in list".
Note: Sys: signifie disquette de lancement.

BindDrivers

Provoque l'installation en mémoire des gestionnaires de périphériques non standard (disque rigide) et les rend ainsi accessibles au système.

SetClock LOAD ;load system time from real time clock (A1000 owners ;should replace the SetClock load with Date)

La commande SETCLOCK avec l'argument LOAD va chercher l'heure de l'horloge de la carte de mémoire des Amiga 500 et 2000 pour ajuster l'horloge interne de l'Amiga. Les ";" servent à ajouter la note avisant le propriétaire d'un Amiga 1000 de remplacer dans ce fichier la commande SETCLOCK par la commande DATE ou par toute commande appropriée pour l'horloge dont il dispose, le cas échéant.

FF >NIL:-0 ;speed up text

La commande FF (Fast Fonts) met en route un programme qui a pour rôle d'accélérer la vitesse d'affichage des textes à l'écran. Le ";" sert à inscrire la note "speed up text".

resident CLI L:Shell-Seg SYSTEM pure add ;activate Shell

Ici la commande RESIDENT rend le "Shell-seg" (situé dans le répertoire L:) résident en l'ajoutant au système (SYSTEM) avec l'argument PURE. Cela signifie que dorénavant le "Shell-Seg" sera résident dans le système et sera toujours prêt à être exécuté sans avoir à être chargé avant. Le ";" sert à ajouter la note "activate Shell".

resident c:Execute pure

La commande EXECUTE située dans le répertoire C est rendue résidente avec l'argument PURE.

mount newcon:

La commande MOUNT rend le périphérique virtuel NEWCON: accessible au système d'exploitation et donc utilisable par le SHELL.

; Le ";" permet de sauter une ligne pour aérer le texte.

failat 11

La commande FAILAT change le niveau de tolérance de l'exécution du fichier de commandes. Ici il est fixé à 11.

run execute s:Startupll ;This lets resident be used for rest of script

La commande RUN crée une seconde tâche qui exécute la commande donnée en paramètre. La commande EXECUTE appelle le fichier de commandes, "Startupll" situé dans le répertoire S, pour en déclencher l'exécution. Le ";" permet d'ajouter la note qui le suit.

wait >NIL: 5 mins ;wait for Startupll to complete (will signal when done)

La commande WAIT demande d'attendre 5 minutes avant de poursuivre l'exécution du fichier "Startup-sequence". Cela laisse le temps au fichier de commandes "Startupll" (voir plus loin) de s'exécuter. A noter que cette opération ne prend pas 5 minutes et que le fichier "Startupll" possède sur sa dernière ligne la commande BREAK qui interrompt la commande WAIT, ce qui relance l'exécution du fichier de commandes "Startup-sequence" aussitôt que celle du fichier "Startupll" est achevée.

; Le ";" permet de sauter une ligne pour aérer le texte.

SYS:System/SetMap usa1 ;Activate the ()/* on keypad

Demande au système d'utiliser le programme "SetMap" situé dans le répertoire "System" pour activer le traducteur des touches "usa1". Ce dernier contient la définition pour les touches de fonctions mathématiques des Amiga 500 et 2000. Le clavier de l'Amiga 1000 est différent, il est donc préférable d'utiliser un autre traducteur des touches. A noter que, si on désire utiliser un autre traducteur, il suffit de remplacer "USA1" par le traducteur de son choix placé au préalable dans le sous-répertoire DEVS/Keymap.

path ram: c: sys:utilities sys:system s: sys:prefs add ;set path for Workbench

La commande PATH avec l'argument ADD demande d'ajouter de nouveaux répertoires dans l'ordre de recherche des commandes. Ici, sont ajoutés dans l'ordre des recherches, le périphérique RAM:, les répertoires C, utilities, system, S, prefs.

LoadWB delay

La commande LOADWB démarre l'Atelier-Workbench et l'argument DELAY demande à la commande d'attendre 3 secondes avant de se terminer.

endcli >Nil:

La dernière ligne contient la suite d'instructions endcli > Nil:. La commande ENDCLI donne l'ordre à l'ordinateur de fermer le CLI en fonction depuis le départ et de le diriger (> est un signe de direction) vers un périphérique fictif (Nil:). Autrement dit, cela signifie de fermer l'écran actif. Cet écran disparu, on a maintenant accès à l'écran créé par la commande LOADWB permettant d'utiliser la souris pour travailler en mode Intuition.

Startupll

Une partie de la séquence de lancement a été déplacée dans un second fichier de commandes appelé "Startupll" de façon à ce que l'exécution de la séquence de lancement (Startup-sequence) soit accélérée par les commandes rendues résidentes. Ce fichier est exécuté après que le "Shell-Seg" a été rendu résident.

Voici ce que signifie chacune des lignes de ce fichier de commandes:

resident c:Resident pure
resident c:List pure ;pre-load LIST and CD
resident c:CD pure
resident c:Mount pure ;the next 3 are loaded for speed during startup
resident c:Assign pure
resident c:Makedir pure
;make IF, ENDIF, ELSE, SKIP, ENDSKIP, and ECHO resident if
;you use scripts much, and can afford the ram.
;also make Failat, WAIT, and ENDCLI resident if you use IconX a lot

A l'aide de la commande RESIDENT et de l'argument PURE les six premières lignes de ce fichier demandent de rendre résidentes les commandes RESIDENT, LIST, CD, MOUNT, ASSIGN, MAKEDIR. Les commandes rendues résidentes seront exécutées presque instantanément parce qu'elles résident en mémoire. Cela évite, par la suite, de nombreuses lectures sur disque et accélère du même coup l'exécution du fichier de commandes. Le ";" sert à insérer des notes pour aider l'utilisateur qui consulte ce fichier.

makedir ram:t

La commande MAKEDIR sert à créer dans le périphérique RAM: un répertoire T: qui servira d'espace de rangement temporaire lors de l'utilisation de certains programmes et pour les fichiers de commandes (éditeurs, compilateurs).

assign T: ram:t ;set up T: directory for scripts

La commande ASSIGN indique au système de modifier le lien avec le périphérique T: et de remplacer cette entité par le T qui vient d'être créé en RAM:. Le ";" sert à écrire la note "Set up T:directory for scripts".

makedir ram:env ;set up ENV: directory

La commande MAKEDIR sert à créer dans le périphérique RAM: un répertoire ENV: qui servira à stocker les variables locales créées par la commande SETENV. Le ";" sert à inscrire la note "Set up ENV: directory".

assign ENV: ram:env

La commande ASSIGN demande de créer un lien entre le répertoire ENV: et le ENV: qui vient d'être créé en RAM:. Ainsi le système saura qu'il faut stocker les variables dans le RAM:env.

makedir ram:clipboards ;set up CLIPS: assign

La commande MAKEDIR sert à créer dans le RAM: un répertoire CLIPBOARDS. Le ";" sert à écrire la note "Set up CLIPS: assign.

assign CLIPS: ram:clipboards

La commande ASSIGN crée un répertoire CLIPS: et l'affecte au répertoire CLIPBOARDS créé en RAM:. CLIPS: sert à stocker dans l'espace mémoire ce qui est copié ou coupé (copy, cut) , plutôt que de l'emmagasiner sur disquette. Cela accélère ces opérations.

```
mount speak: ;Just mounting doesn't take much ram at all
mount aux:
mount pipe:
```

Ces trois lignes de commandes utilisent la commande MOUNT pour rendre accessibles au système les trois périphériques virtuels SPEAK:, AUX:, PIPES. La note qui suit le ";" indique que l'installation d'un périphérique avec la commande MOUNT nécessite peu d'espace mémoire.

```
resident Mount remove ;If you have enough ram, keep these resident
resident Assign remove  ;by removing these lines
resident Makedir remove
```

La commande RESIDENT, avec l'argument REMOVE utilisé sur ces trois lignes de commmandes, a pour fonction de retirer de la liste des commandes résidentes les commandes MOUNT, ASSIGN et MAKEDIR. La note qui suit le ";" indique que, si l'espace mémoire est suffisant, on a avantage a conserver ces commandes résidentes, et, pour ce faire, on élimine ces trois lignes.

; Le ";" permet de sauter une ligne pour aérer le texte.

```
break 1 C ;signal to other process its ok to finish
```

La commande BREAK a pour fonction d'interrompre la commande WAIT 5 minutes du fichier de commandes "Startup-sequence" de la tâche 1 et de permettre ainsi la poursuite du déroulement du premier fichier de commandes.

La séquence de lancement, sa flexibilité et ses avantages

L'examen des fichiers de commandes permet de comprendre que l'un des rôles de la séquence de lancement est de permettre à l'utilisateur de programmer des opérations répétitives et ainsi éviter d'avoir à répéter une série d'opérations fastidieuses toutes les fois qu'il utilise son ordinateur.

Exemples:

Pour charger automatiquement le traducteur des touches de son choix, on n'a qu'à inclure, dans la séquence de lancement, la ligne d'instructions suivante:

system/SetMap (nom-du-clavier)

Il est possible que cette ligne existe déjà. Dans un tel cas, il suffit de changer le nom du traducteur. Cela implique que le sous-répertoire Devs/Keymap doit contenir ledit clavier. A noter que Commodore a placé différents traducteurs sur la disquette Extras 1.3. Il peut donc être nécessaire de déplacer le traducteur désiré de la disquette Extras 1.3 dans le sous-répertoire Devs/keymap de la disquette Atelier-Workbench.

Si on veut que soit créée une icône de disquette RAM:, on ajoute la commande suivante:

DIR ram:

Si on veut que soient affichés des messages lors du lancement, on utilise la commande ECHO et, entre guillemets, on insère le message.

Si on désire qu'un détecteur de virus, du genre VirusX qui se trouve sur l'une des disquettes Outils, prévienne automatiquement lorsqu'une disquette suspecte est introduite dans un des lecteurs à disquette, on copie le programme VirusX dans le répertoire courant et on ajoute la ligne d'instruction suivante:

VirusX

ATTENTION! Dans la séquence de lancement de l'Atelier-Workbench 1.3, cette ligne d'instruction doit précéder la commande **CD c:** à moins d'avoir transféré le programme VirusX dans le répertoire c:.

Si on veut que l'écran soit d'une autre couleur lors du lancement, on indique ses préférences par des commandes insérées dans les lignes d'instruction, et le tour est joué.

Amorce ultra-rapide

La séquence de lancement la plus rapide et aussi la plus simple consiste à créer une séquence contenant simplement un mot, soit celui du fichier qu'on désire utiliser. Par exemple, pour lancer immédiatement le programme "Textcraft" situé sur la disquette de lancement, il suffit simplement d'écrire le mot "Textcraft" dans la séquence de lancement. Rapidement l'ordinateur recherchera et lancera ce programme. C'est intéressant, mais attention, si on quitte le programme, on se retrouve devant un écran presque vide présentant un signal de sollicitation, car le programme "LoadWB" n'a pas été chargé lors du lancement. Ce signal invite à utiliser le mode par commandes.

Pour avoir accès au mode Intuition, il suffit de taper la commande LOADWB, de faire RETOUR, puis de taper la commande ENDCLI et faire RETOUR.

Cette même procédure s'applique lorsqu'on désire revenir au mode Intuition après avoir court-circuité la séquence de lancement (cela se fait en appuyant simultanément sur les touches CTRL et D au moment voulu).

Amorce lente

Il va de soi qu'une séquence de lancement comportant des commandes complexes s'exécute avec plus de lenteur, mais elle a l'avantage de créer le cadre de travail que désire l'utilisateur.

Comprendre

L'une des premières choses que fait l'Amiga quand on introduit la disquette Atelier-Workbench est d'aller lire la séquence de lancement (Startup-sequence).

La séquence de lancement est simplement une liste ordonnée de commandes à exécuter par l'ordinateur. L'exécution de ces commandes est soumise à l'ordre des instructions contenues dans un texte écrit en ASCII. Il s'agit d'ordonner une suite de commandes en utilisant un langage de programmation simple mais puissant. Les commandes contenues dans la séquence de lancement sont un mélange de commandes provenant du répertoire C et de tout autre répertoire défini explicitement.

La séquence de lancement permet à l'utilisateur de programmer le lancement de l'appareil selon ses besoins. Cela permet, entre autres, le lancement de programmes, le transfert de programmes et de commandes les plus usuelles de l'AmigaDos dans le disque virtuel RAM:, le RAD: ou le VD0:, etc. Cela permet également de rendre des commandes résidentes. Le fichier de la séquence de lancement doit faire partie du répertoire S:.

LES TURMITES (III)

Nous attaquons cette fois la gestion des affichages, les initialisations et le scrolling. Insistons sur le fait que les deux précédents articles (ST Mag 40 et 41) sont indispensables pour la bonne marche de la programmation. D'une façon générale, le manque de place nous a obligés à découper ce "gros" listing en plusieurs parties, mais les plus impatients d'entre vous peuvent se procurer la disquette "Amiga" à la Boutique de Pressimage qui contient, depuis notre premier épisode, l'ensemble du programme ainsi que les autres listings Amiga déjà publiés pour tout ou partie dans le journal.

François Fleuret

```
;*************************************************
;*   Routine d'affichage des pavés de nourriture *
;*************************************************

Paves:
 move.w Xmousew,d0; si on dépasse à droite, alors on corrige
 cmp.w #498,d0
 bmi PasPavesDroite
 move.w #497,d0
PasPavesDroite:
 move.w d0,d4
 and.w #$0f,d4          ; masque pour le décalage
 move.w Ymousew,d1
 cmp.w #497,d1          ; teste le dépassement en bas
 bmi PasPavesBas
 move.w #496,d1
PasPavesBas:
                        ; calcul de l'adresse en mémoire vidéo
 lsl.w #6,d1            ; 64 octets par ligne (512 pts)
 lsr.w #3,d0            ; on divise X par 8
 and.w #$fffe,d0        ; force une adresse paire
 add.w d1,d0            ; X+Y
 move.l plan1,a0        ; + base de la mémoire
 lea (a0,d0),a0
 move.l plan2,a1        ; idem plan 2
 lea (a1,d0),a1

 move.l #$ffff0000,d0 ; dessin du bloc
 lsr.l d4,d0            ; avec décalage
 not.l d0               ; et masque

 move.w Color,d1
 clr.l d2               ; dessin pour le plan 1
 clr.l d3               ; dessin pour le plan 2
; on modifie d2 et d3 en
; fonction des bits de la couleur
 btst #0,d1
 beq PasPlan0
 move.l d0,d2
 not.l d2
PasPlan0:
 btst #1,d1
 beq PasPlan1
 move.l d0,d3
 not.l d3
PasPlan1:
 move.w #15,d4          ; blocs de 16 lignes de haut
BouclePaves:
 and.l d0,(a0)
 or.l d2,(a0)
 add.l #64,a0
 and.l d0,(a1)
 or.l d3,(a1)
 add.l #64,a1
 dbf d4,BouclePaves
FinPaves:
 rts
;*********************************
; Routine d'affichage d'un 'Cerveau'
;*********************************
AfficheCerveau:
 movem.l d0-a6,-(sp)
 tst.l AdrSelect        ; y a-t-il une turmite sélectionnée ?
 beq PasCerveau         ; si non, alors ne rien faire
 move.l SpriteTableau,a0  ; sprite pour afficher le tableau
 lea 4(a0),a0           ; adresse des données bitmap du sprite
 move.l AdrSelect,a1    ; adresse de la turmite sélectionnée
 lea Table(a1),a1   ; adresse de son tableau de comportement
 move.w #3,d0           ; 4 états
BoucleEtat:
 move.w #3,d1           ; et 4 couleurs par état
BoucleCouleur:
 move.b (a1)+,d3
 add.b #1,d3
 ext.w d3
 lsl.w #4,d3
 lea TableCarDirections,a2  ; table des caractères (g/a/d)
 lea (a2,d3.w),a2
 move.b (a1)+,d3
 ext.w d3
 lsl.w #4,d3
 lea TableCarEtats,a3 ; table des caractères états (A/B/C/D)
 lea (a3,d3.w),a3
 move.b (a1)+,d3
 ext.w d3
 lsl.w #4,d3
 lea TableCarChiffres,a4   ; chiffres (0/1/2/3...)
 lea (a4,d3.w),a4
  move.w #5,d2
BoucleCar:
```

```
 moveq #0,d3
 or.w (a2)+,d3
 or.w (a3)+,d3
 or.w (a4)+,d3
 move.w d3,(a0)+
 clr.w (a0)+
 dbf d2,BoucleCar
 add.w #256-24,a0
 dbf d1,BoucleCouleur
 add.w #28-256*4,a0
 dbf d0,BoucleEtat

PasCerveau:
 movem.l (sp)+,d0-a6
 rts

ClearTableau:
;vide la bitmap du sprite tableau
 movem.l d0/a0,-(sp)
 move.l SpriteTableau,a0
 lea 4(a0),a0
 move.w #55,d0
BoucleClearTableau:
 clr.w 256(a0)
 clr.w 512(a0)
 clr.w 768(a0)
 clr.w (a0)+
 dbf d0,BoucleClearTableau
 movem.l (sp)+,d0/a0
 rts
; ********************
; Affichage nombre de
; turmites à l'écran
; ********************
AfficheNombre:
 move.w nombre,d0
; table caractères numériques
 lea TableCarChiffres,a4
; utilise le reste d'une
; division par 10
 divu #10,d0
 swap d0
 lsl.w #4,d0
 lea (a4,d0.w),a1
 clr.w d0
 swap d0
 divu #10,d0
 swap d0
 lsl.w #4,d0
 lea (a4,d0.w),a2
 clr.w d0
 swap d0
 divu #10,d0
 swap d0
 lsl.w #4,d0
 lea (a4,d0.w),a3
 clr.w d0
 swap d0
 divu #10,d0
 swap d0
 lsl.w #4,d0
 lea (a4,d0.w),a4
; adresse datas de ce sprite
 move.l SpriteNombre,a0
 lea 4(a0),a0  ; datas bitmap
; caractères de 6 lignes
 move.w #5,d0
BoucleAfficheNombre:
; recopie les caractères de
; la table dans le sprite.
 move.w (a4)+,d1
 lsl.w #5,d1
 or.w (a3)+,d1
 move.w d1,(a0)
 clr.w 2(a0)
 move.w (a2)+,d1
 lsl.w #5,d1
 or.w (a1)+,d1
 lsl.w #6,d1
 move.w d1,32(a0)
 clr.w 34(a0)
 add.l #4,a0
 dbf d0,BoucleAfficheNombre
 rts

;******************************
;Routine de traçage du terrain
;******************************

; D0=Largeur du côté
; (32/64/128/256)

DrawWorld:
 movem.l d0-a6,-(sp)
; bases des bitplans
 move.l plan1,a0
 move.l plan2,a1
 moveq #0,d1
 moveq #0,d2
 move.w #$1fff,d3
; tout d'abord, on efface
ClearScreen:
 move.l d1,(a0)+
 move.l d2,(a1)+
 dbf d3,ClearScreen

; calcul adresse de début,
; on passe des octets pour
; la marge de gauche
 move.w d0,d1
 move.w d0,d2
 lsl.w #5,d1
 lsr.w #4,d2
 add.w d2,d1
 move.l plan1,a0
 move.l plan2,a1
 add.w d1,a0
 add.w d1,a1

 bsr ChoixBits  ; voir ci-dessous
 move.l #512,d3 ; combien de carrés ?
 divu d0,d3
; un DIVU ne peut pas faire
; de mal (enfin, pas trop...)
 lsr.w #1,d3
 sub.w #1,d3
 move.w d3,a2

 move.w d0,d3
 sub.w #1,d3
 move.w d3,a3

 move.w d0,d6
 lsr.w #4,d6
 sub.w #1,d6
 lsl.w #6,d0

 move.w a2,d4
; boucle verticale des carrés
DWboucleV:
 move.w a2,d5
; idem horizontale
DWboucleH:

 move.w a3,d3
; les boucles pour UN carré
DWpetiteboucleV:
 move.w d6,d7
DWpetiteboucleHfill:
 move.w d1,(a0)+
 move.w d2,(a1)+
 dbf d7,DWpetiteboucleHfill
 move.w d6,d7
DWpetiteboucleHclear:
 clr.w (a0)+
 clr.w (a1)+
 dbf d7,DWpetiteboucleHclear
 dbf d3,DWpetiteboucleV

 dbf d5,DWboucleH
 add.w d0,a0
 add.w d0,a1
 dbf d4,DWboucleV
 movem.l (sp)+,d0-a6
 rts

; remplit l'écran avec
; la couleur en cours
ClearWorld:
 movem.l d0-a6,-(sp)
 move.l plan1,a0
 move.l plan2,a1
 bsr ChoixBits
 move.w #$3fff,d3
ClearScreen2:
 move.w d1,(a0)+
 move.w d2,(a1)+
 dbf d3,ClearScreen2
 movem.l (sp)+,d0-a6
 rts

; la routine qui détermine
; les mots en fonction des
; bits de la couleur
ChoixBits:
 moveq #0,d1
```

```
 moveq #0,d2
 move.w Color,d3
 btst #0,d3
 beq PasCls0
 move.w #-1,d1
PasCls0:
 btst #1,d3
 beq PasCls1
 move.w #-1,d2
PasCls1:
 rts
;************************
; *** Initialisations ***
;************************
Inits:
 lea $dff000,a6

 move.w #$7fff,$9a(a6)    ; INTENA
 move.w #$7fff,$96(a6)    ; DMACON
 move.w #$c020,$9a(a6)    ; IT vbl uniquement
 move.w #$83a0,$96(a6)    ; copper et bitplan

 move.l SpriteVide,a0
 clr.l (a0)+
 clr.l (a0)+

 move.l SpriteTableau,a0  ; positionne les sprites
 move.w #$2e50,(a0)
 move.w #$4a00,2(a0)
 move.w #$2e58,256(a0)
 move.w #$4a00,256+2(a0)
 move.w #$2e60,512(a0)
 move.w #$4a00,512+2(a0)
 move.w #$2e68,768(a0)
 move.w #$4a00,768+2(a0)

 move.l SpriteNombre,a0
 move.w #$2ec2,(a0)
 move.w #$3400,2(a0)
 move.w #$2eca,32(a0)
 move.w #$3400,34(a0)

 move.l SpriteCursor,a0
 lea ImageCursor,a1
 bsr InitSprites
 move.l SpriteSelect,a0
 lea ImageSelect,a1
 bsr InitSprites

 move.l copper_list,a0

 move.l #$00e00000,(a0)+  ; la copper-list met les pointeurs
 move.l #$00e20000,(a0)+  ; bitplans
 move.l #$00e40000,(a0)+
 move.l #$00e60000,(a0)+
 move.l #$01020000,(a0)+
 move.l #$fffffffe,(a0)+

; On se met en 256*256 2 bitplanes (4 coul.) avec 16 points
; cachés à gauche pour faire un scrolling avec le delay.
 move.l copper_list,$80(a6)  ; COP1LC
 clr.w $88(a6)               ; COPJMP1
```

```
 move.l #$2ca12ca1,$8e(a6)   ; DIWSTRT et DIWSTOP
 move.w #$2200,$100(a6)      ; BPLCON0
 clr.l $102(a6)              ; BPLCON1 et BPLCON2
 move.w #30,$108(a6)         ; BPLMOD1
 move.w #30,$10a(a6)         ; BPLMOD2
 move.l #$004000c0,$92(a6)   ; DDFSTRT et DDFSTOP
 move.l #irq,$6c             ; vecteur irq

 move.w #$0024,$dff104       ; priorité des sprites
 move.w #$555,$dff180+17*2   ; couleurs de ces mêmes sprites
 move.w #$00f,$dff180+18*2
 move.w #$fff,$dff180+19*2
 move.w #$fff,$dff180+21*2
 move.w #$fff,$dff180+25*2
 move.w #$fff,$dff180+29*2

 clr.w $dff180               ; couleur playfield
 move.w #$f00,$dff182        ; noir, rouge, bleu, jaune
 move.w #$00f,$dff184
 move.w #$ff0,$dff186
 bsr InitClavier             ; initialise le clavier
 rts

InitSprites
 move.w #15,d0
 clr.l (a0)+
RecopieSprites:
 move.l (a1)+,(a0)+
 dbf d0,RecopieSprites
 clr.l (a0)+
 rts
;******************************
;*** Routine du 'scrolling' ***
;******************************
Screen:
 tst.w FlagSuivre      ; est-on-en mode suivi automatique ?
 beq PasSuivie         ; non
 tst.l AdrSelect       ; une turmite est-elle sélectionnée ?
 beq PasSuivie         ; non
 move.l AdrSelect,a0   ; si oui
 tst.w V(a0)           ; est-elle vivante ?
 beq PasSuivie         ; non
PasFinSuivre:
 move.w X(a0),d0       ; prend ses coordonnées
 sub.w #128,d0         ; centre
 move.w d0,Xwindow     ; et les met dans celles de la fenêtre
 move.w Y(a0),d0
 sub.w #128,d0
 move.w d0,Ywindow
PasSuivie:

 move.w Xwindow,d0     ; on corrige éventuellement les
 bpl PasWindowGauche   ; coordonnées de la fenêtre
 moveq #0,d0
 move.w d0,Xwindow
 bra PasWindowDroite
PasWindowGauche:
 cmp.w #256,d0
 bmi PasWindowDroite
 move.w #256,d0
 move.w d0,Xwindow
PasWindowDroite:
```

```
 move.w Ywindow,d1
 bpl PasWindowHaut
 moveq #0,d1
 move.w d1,Ywindow
 bra PasWindowBas
PasWindowHaut:
 cmp.w #256,d1
 bmi PasWindowBas
 move.w #256,d1
 move.w d1,Ywindow
PasWindowBas:

 move.w Xmouse,Xmousew      ; coordonnées absolues de
 add.w d0,Xmousew           ; la souris dans le terrain
 move.w Ymouse,Ymousew
 add.w d1,Ymousew

 lsl.w #6,d1          ; calcul adresse de départ des bitplans
 move.w d0,d2
 lsr.w #3,d2
 and.w #$fffe,d2
 add.w d2,d1
 not.w d0
 and.w #15,d0
 moveq #0,d2
 move.w d0,d2
 lsl.w #4,d0
 or.w d0,d2

 move.l copper_list,a1     ; met le delay dans la copper-list
 move.w d2,2+16(a1)

 move.l plan1,a0
 add.w d1,a0
 move.l a0,d0

 move.w d0,6(a1)          ; met l'adresse dans la copper-list
 swap d0
 move.w d0,2(a1)
 swap d0
   add.l #$8000,d0        ; deuxième bitplan
 move.w d0,6+8(a1)
 swap d0
 move.w d0,2+8(a1)

 moveq #0,d4        ; gestion du sprite curseur de sélection
 moveq #0,d5
 move.l AdrSelect,d0
 beq PasSpriteSelection
 move.l d0,a0
 tst.w V(a0)
 beq PasSpriteSelection
 move.w X(a0),d0
 sub.w Xwindow,d0
 bmi PasSpriteSelection
 cmp.w #255,d0
 bpl PasSpriteSelection
 move.w Y(a0),d1
 sub.w Ywindow,d1
 bmi PasSpriteSelection
 cmp.w #255,d1
 bpl PasSpriteSelection
```

```
 add.w #$9d,d0          ; pareil que pour celui de la souris
 add.w #$2a,d1

 move.w d1,d2
 add.w #8,d2

 move.w d0,d3
 lsr.w #1,d3
 move.w d1,d4
 lsl.w #8,d4
 move.b d3,d4

 move.b d2,d5
 lsl.w #8,d5
 roxl.w #8,d1
 roxl.b #1,d5
 roxl.w #8,d2
 roxl.b #1,d5
 roxr.b #1,d0
 roxl.b #1,d5

PasSpriteSelection:
 move.l SpriteSelect,a0
 move.w d4,(a0)+
 move.w d5,(a0)

 rts
;*************************
;*** Routines de l'IRQ ***
;*************************
irq:
 movem.l d0-a6,-(sp)
 bsr GereSprites
 bsr Souris
 bsr Screen
 move.w #-1,VblFlag
 movem.l (sp)+,d0-a6
 move.w #$20,$dff09c        ; it vbl traitée INTREQ
 rte
;*****************************************
; Routine de gestion des sprites (début VBL)
;*****************************************
GereSprites:
 move.l SpriteCursor,$dff120 ; SPR0PT
 move.l SpriteSelect,$dff124 ; SPR1PT
 move.l SpriteNombre,d0
 move.l d0,$dff128           ; SPR2PT
 add.l #32,d0
 move.l d0,$dff12c           ; SPR3PT
 move.l SpriteTableau,d0     ; 4 sprites pour le tableau
 move.l d0,$dff130           ; SPR4PT
 add.l #256,d0
 move.l d0,$dff134           ; SPR5PT
 add.l #256,d0
 move.l d0,$dff138           ; SPR6PT
 add.l #256,d0
 move.l d0,$dff13c           ; SPR7PT
 rts
```

Ouf ! Plus qu'une partie, et le mois prochain, avec la gestion des entrées (souris, clavier), nous en aurons terminé ! Bon courage...

NEWS MAC

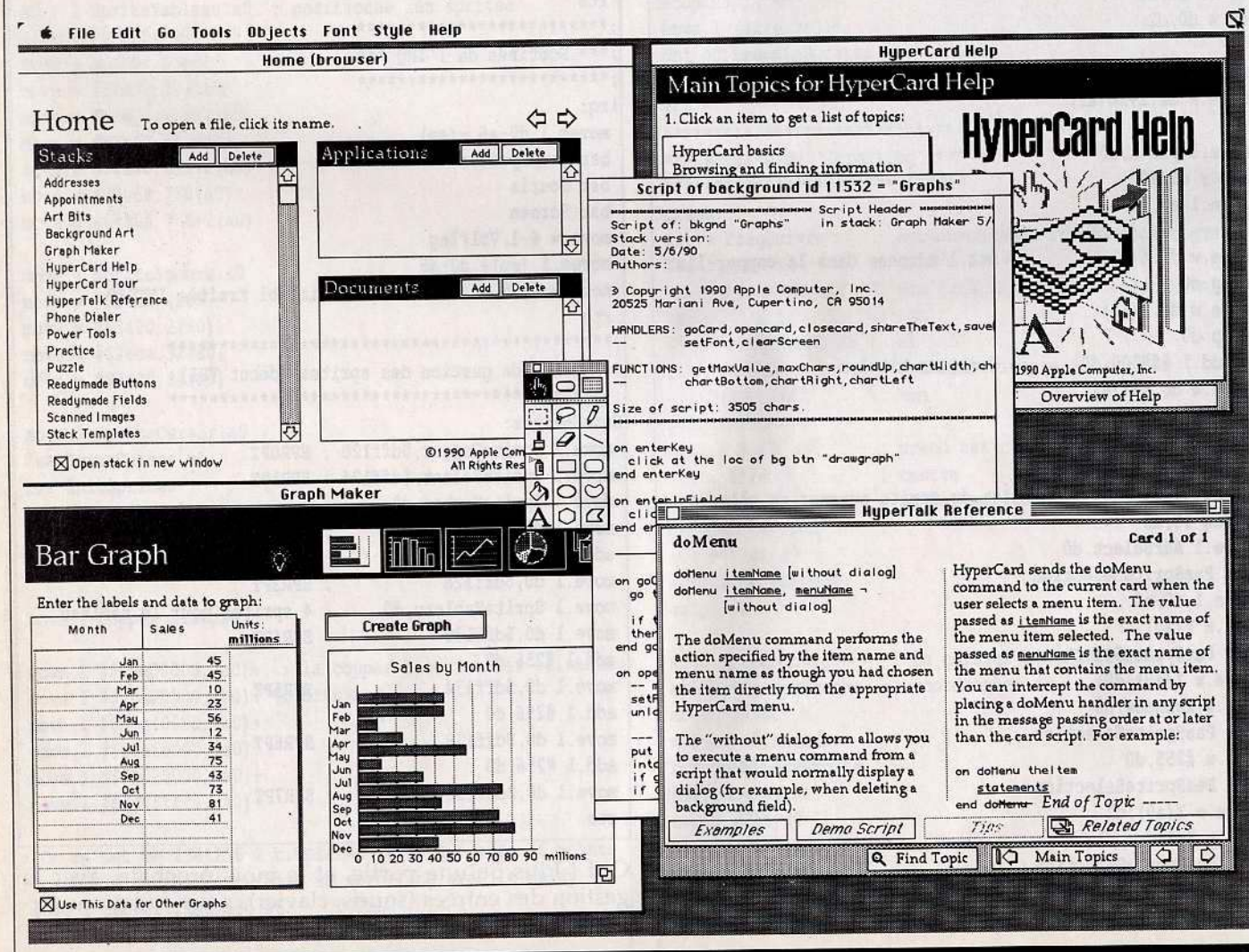
Ce mois-ci, les news Mac sont entièrement consacrées à ce qui se passe chez Apple. Et il s'en passe, des choses, chez eux (commentaire tout à fait innocent vis-à-vis d'autres sociétés, bien sûr...).

ET VOILA HYPERCARD 2 !

Il n'est plus nécessaire de vous présenter HyperCard, je l'espère. Pour ceux qui se seraient endormis ces dernières années, rappelons tout de même qu'il s'agit d'un programme créé par Bill Atkinson (celui-là même qui fit MacPaint aux débuts du Macintosh), permettant la création simple d'applications, entièrement à la souris, avec la possibilité d'écrire ensuite des "scripts" complets allant un peu plus loin que les options standard.

Ce programme, qui permet de mettre en oeuvre relativement facilement les concepts d'hypertexte, par la possibilité d'établir des liens entre un objet d'une "carte" et une autre carte, a connu un vif succès, d'autant plus que son langage de programmation, assez original puisqu'il essayait de suivre le plus possible la syntaxe de la langue anglaise, a permis la création de très nombreuses applications plus ou moins complexes. Je vous invite évidemment à vous reporter à notre série sur le sujet, dans les numéros 35 à 41.

Mais trêve de bavardages, voyons ce qu'apporte de nouveau la version 2 – qui sera disponible en France au mois d'octobre – et qui est évidemment une mouture complètement nouvelle, rien à voir avec celles qui ont suivi la 1.0, et qui n'apportaient que des changements mineurs. Tout d'abord, chaque pile peut avoir une taille de son



choix, chacune des dimensions étant limitée à 1280 pixels. Ensuite, on peut ouvrir simultanément plusieurs piles qui seront affichées chacune dans une fenêtre, et ce, à concurrence de la mémoire disponible. Au passage, notez qu'HyperCard 2 exige déjà le système 6.0.4, on se demande quelle version il réclamera à sa sortie !

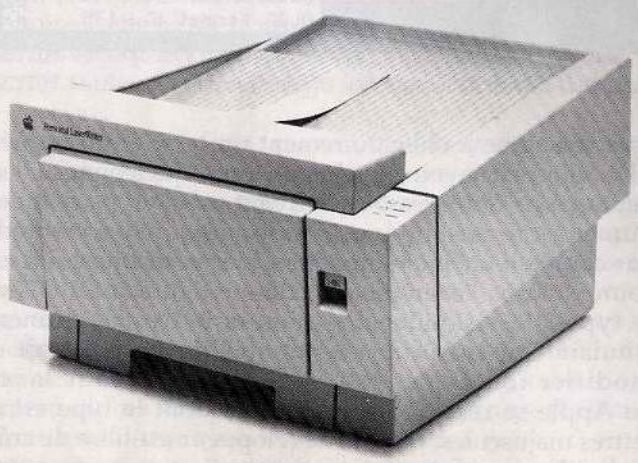
Par contre, pas de gestion de la couleur à l'horizon. Le vectoriel ne semble pas non plus avoir été honoré, du moins à première vue (nous n'avons pu faire qu'un tour très rapide de la chose, mais vous trouverez bien sûr plus de détails dans notre prochain numéro).

Continuons notre rapide tour par les aspects de programmation. HyperTalk, le langage, a reçu des améliorations, mais ce qui frappe le plus, c'est la présence de nombreux outils pour en faciliter l'usage : on a maintenant un véritable éditeur à sa disposition, chaque script ayant sa propre fenêtre, et plusieurs scripts pouvant évidemment être ouverts simultanément. Un débugger a même été ajouté, permettant de tracer les handlers, de visualiser le contenu des variables, d'être informé des messages, etc. Bref, un énorme changement, qui en cache encore beaucoup d'autres.

Je vous invite donc à nous rejoindre le mois prochain pour plus de détails.

PLUS LENT ? MOINS CHER !

Apple a récemment présenté sa nouvelle gamme d'imprimantes laser, les "Personal Laserwriters". Basées sur des mécaniques Canon LX, elles sont plus compactes (avec tout de même 2 bacs à papier), un peu plus lentes (4 pages par minute au lieu des 8 habituelles), mais surtout moins chères, et elles risquent d'en séduire plus d'un.
On trouve tout d'abord la Personal Laserwriter SC, qui pour moins de 13000F, offre tout de même un 68000 à 8 MHz, 1 Mo de RAM, et se connecte à un Macintosh (et un seul) par le port SCSI, n'acceptant que QuickDraw (mais avec ATM et les fontes TrueType du Système 7, c'est loin d'être nécessaire dans la plupart des cas).



La Personal Laserwriter NT, quant à elle, dispose du même processeur, mais cadencé cette fois à 12 MHz, et accompagné de 2 Mo de RAM, extensibles à 8, permettant de faire fonctionner PostScript qui est intégré, ainsi que 37 polices de caractères. Le tout est accessible via LocalTalk, permettant le partage de ladite imprimante sur un réseau.
Bref, de bien belles imprimantes pour qui n'a pas besoin de la vitesse ou de l'encombrement d'une LaserWriter II. Notez toutefois que si la Personal LaserWriter SC peut être transformée en son homologue NT, il n'est évidemment pas possible de transformer le tout en LaserWriter II, évidemment !

JDD 90, SYSTEME 7 & CO...

Nous vous parlions la dernière fois des Journées du Développement '90 organisées par Apple fin juin. Le thème principal en a été la version 7 du Système, et les développeurs présents ont reçu un magnifique CD-ROM le contenant, ainsi que quelques autres "bricoles" comme une version préliminaire sur disque du volume VI d'Inside Mac, consacré à la version sus-citée du dit Système.
Evidemment, il était à la charge de chacun de trouver un lecteur de CD-ROM (avec en plus un semblant de rupture de stock sur cet article chez Apple...). Les chanceux ou les débrouillards ont dont pu découvrir une "alpha-version" du Système 7, qui est encore très lente, mais qui a le mérite de tourner, et de prouver que tout ceci est autre chose qu'un beau discours des dirigeants d'Apple.
Ainsi, les développeurs français vont maintenant pouvoir travailler en connaissant les caractéristiques précises du Système 7, en tirer parti dans leurs applications, éventuellement corriger ce qui pourrait désormais poser problème avec cette nouvelle version.
L'un des plus gros apports du Système 7, au-delà des aspects extérieurs comme le nouveau Finder, est la présence de l'"Edition Manager", un nouvel ensemble de routines du Système permettant de gérer le partage de données entre applications. Il suffira dans les applications qui le prévoieront, de sélectionner un objet, et de le rendre "public", pour pouvoir ensuite y "souscrire" dans un autre document, toutes les modifications dans le premier fichier étant alors automatiquement reportées dans la partie correspondante du deuxième.
Encore un peu de patience, ça arrive, on vous en reparlera bien sûr...

LES PRIX BAISSENT

Plein de bonnes nouvelles pour ceux qui voudraient un Mac, puisqu'on trouve des Mac Plus et SE à des prix défiant toute concurrence (jusqu'à 5500F pour un Mac Plus tout nu, ou 10000F pour un Mac SE avec disque dur de 40 Mo!), pour ceux aussi qui ont déjà un Mac II, et qui vont pouvoir le transformer en IIfx pour 15000 petits francs (auxquels il faut rajouter le prix de la RAM, celles-ci étant différents). Sachant que le Mac II tournait aux alentours de 20000F d'occasion ces derniers temps, quelques-uns vont avoir un IIfx pour pas cher !

LES RESSOURCES DU MACINTOSH

HISTORIQUE

Le catalogue des fichiers

Avant la génération Macintosh, en 1984, l'antiquité informatique connaissait déjà les fichiers, mais avait beaucoup de mal à les classer. On comptait sur l'utilisateur pour fournir un nom auquel le système d'exploitation ajoutait gracieusement une extension pour indiquer de quel type de fichier il pouvait s'agir. Par rapport aux systèmes basés sur ce principe comme Unix ou MS-DOS, Apple avait déjà innové avec l'Apple][en pensant à stocker le type du fichier dans un octet spécial du catalogue, au lieu de l'inclure dans le nom du fichier.

Mais dans le cas du Macintosh, un seul octet était nettement insuffisant pour pouvoir contenir toutes les informations qu'il était prévu de faire figurer au catalogue : il fallait au moins 4 octets pour le type du fichier, encore 4 octets pour l'application qui l'avait créé, sans compter les 256 octets nécessaires au codage de l'icône apparaissant sur le bureau. L'idée géniale vînt alors : pourquoi ne pas ajouter à chaque fichier un deuxième fichier qui contiendrait toutes les informations annexes dont il pourrait avoir besoin ? Les ressources étaient nées.

Les paramètres de la toolbox

Bien sûr, il y avait d'autres raisons pour créer une partie "ressources" associée à chaque fichier. Les applications qui devaient toutes utiliser la boîte à outils passaient une grande partie de leur temps à fournir à ces routines des structures de données compliquées, décrivant la forme d'une fenêtre, les éléments d'un dialogue, ou encore la barre de menu. Toutes les techniques utilisées auparavant consistaient à coder dans le programme source ce genre de données. Ce procédé est conceptuellement très mauvais, mais on ne voyait pas comment faire autrement. Encore une fois, l'idée d'ajouter un fichier associé au programme et qui contiendrait toutes ces structures s'imposa chez Apple. Il suffirait alors au programme de fournir aux routines de la boîte à outils la référence de la structure que l'on voulait utiliser : la toolbox se débrouillerait ensuite pour obtenir la description exacte dans le fichier des ressources. Ainsi, plus aucun problème pour traduire un logiciel : il suffira de modifier le texte grâce à un éditeur de ressources, sans qu'il faille recompiler quoi que ce soit.

CRÉATION ET UTILISATION DE RESSOURCES

Mode d'emploi général de ResEdit

ResEdit est cet éditeur de ressource, développé par Apple. Après l'avoir lancé, il commence par proposer la liste des fichiers présents sur le disque. Lorsqu'on en a choisi un, il affiche ensuite la liste des types de ressources que contient ce fichier. Enfin, lorsqu'on sélectionne un type donné, il fournit la liste des ressources correspondantes, avec leur numéro. Il est alors possible d'éditer et de modifier individuellement chaque ressource.

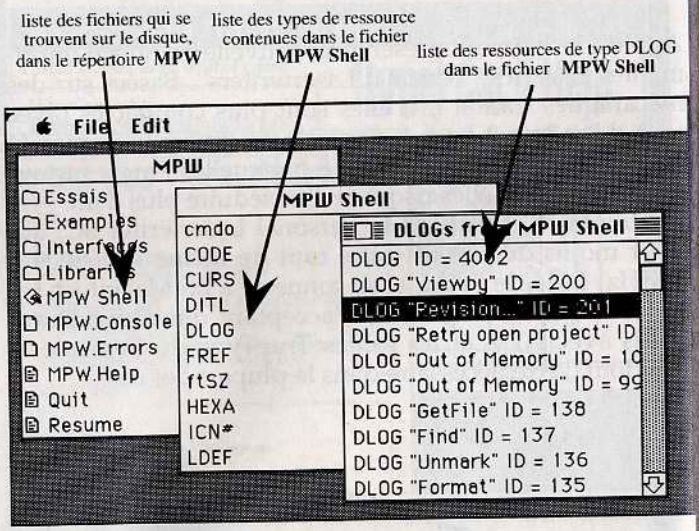


Figure 1 - ResEdit

Une ressource a obligatoirement un type et un numéro (ID), elle peut avoir en plus un nom quelconque. Il ne peut exister deux ressources ayant même type et même numéro, car ce sont avec ces données qu'on y accède. Par contre, le même nom peut servir pour plusieurs ressources. Dans l'exemple sélectionné figure 1, **DLOG** est le type, "Revision." est le nom et 201 est le numéro. Choisir **Get Info** dans le menu **File** permettrait de modifier toutes ces caractéristiques. Il faut savoir qu'Apple se réserve les ressources dont le type est en lettres majuscules, mais le développeur est libre de créer et d'utiliser ses formats personnels.

La gestion des ressources

Somme toute, chaque ressource n'est qu'un ensemble d'octets auquel l'on peut accéder par le biais d'un type et d'une ID. Cependant selon le type de la ressource, les octets ont des significations variées et à condition que ResEdit les connaisse, il peut présenter la ressource de manière plus lisible. Ainsi, sur la figure 2 un amateur d'hexadécimal pourrait décoder la ressource à la main, mais ResEdit s'en charge aussi bien pour l'humain normal.

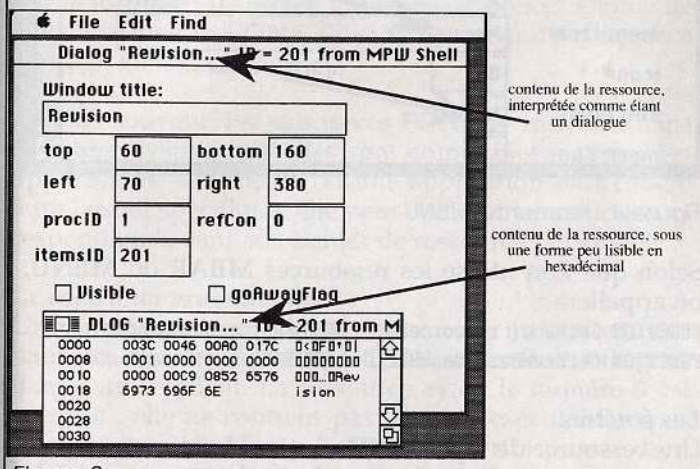


Figure 2 - une ressource

Concernant la programmation, les appels aux routines seront donnés en Pascal, car c'est le langage utilisé par Apple dans Inside Macintosh. Par exemple, le nom d'une ressource est défini par :

```
TYPE ResType = PACKED ARRAY[1..4] OF CHAR ;
```

Lorsqu'on demande au Resource Manager de charger une ressource, celui-ci la cherche dans tous les fichiers qui ont été ouverts en commençant par le plus récent. Si l'application *toto* a ouvert le document *titi*, la ressource sera d'abord recherchée parmi celles de *titi*, puis celles de *toto*, pour finir par celles du fichier System, qui a été ouvert au démarrage de l'ordinateur. Ce procédé est excellent pour hiérarchiser les utilisations des ressources : un document peut avoir ses propres ressources pour lui tout seul et une application peut en avoir pour tous ses documents. Pour charger une ressource quelconque en mémoire si elle ne s'y trouve pas déjà, on peut utiliser :

```
FUNCTION GetResource(theType:ResType;theID:INTEGER):Handle
FUNCTION GetNamedResource(theType:ResType;name:Str255):Handle
```

A ce propos, il faut savoir que la mémoire sur Macintosh est partagée entre plusieurs blocs de tailles modulables : un bloc pour le système appelé le "system heap", et un bloc par application : le "application heap". Ces blocs contiennent en fait un index des pointeurs et handles alloués. L'action de charger une ressource correspond à allouer un handle de la taille judicieuse et à y placer le contenu de la ressource. Si l'on veut créer une nouvelle ressource d'un type donné, il faut fournir un numéro qui ne soit pas déjà alloué. Pour en trouver un, la fonction suivante accomplit des merveilles :

```
FUNCTION UniqueID(theType:ResType):INTEGER ;
```

On pourra enfin créer la ressource en utilisant :

```
PROCEDURE AddResource( theData: Handle; theType: ResType ;
theID: INTEGER ; name: Str255 ) ;
```

Mais le plus souvent, un programme veut modifier une ressource déjà existante. Pour cela, il faut d'abord la charger en mémoire par un **GetResource**, puis appeler successivement les deux procédures :

```
PROCEDURE ChangedResource( theResource : Handle ) ;
PROCEDURE WriteResource(theResource : Handle ) ;
```

Si l'on veut pouvoir manipuler la partie ressource comme un fichier normal, il existe un appel du File Manager qui le permet :

```
FUNCTION OpenRF( filename : Str255 ; vRefNum : INTEGER ;
VAR refNum : INTEGER ) : OSErr ;
```

Pour copier un fichier entier, il suffira de copier d'abord la partie data, qui s'obtient de façon standard par un **FSOpen**, puis de copier la partie ressource grâce à **OpenRF**. Ceci explique que lors de la transmission par un programme standard (par exemple écrit en C utilisant stdio.h), il manque le plus souvent une importante partie du fichier : seule la partie data a été transmise.

LES TYPES DE RESSOURCE CELEBRES

Les dialogues

Une boîte de dialogue, c'est une fenêtre qui sert à demander des informations à l'utilisateur. Il y a deux types d'information à fournir pour en créer une : d'une part, il faut indiquer les dimensions et le type de la fenêtre, ce que font les ressources **DLOG**. Il faut d'autre part indiquer la liste des items, qui se trouvent dans les ressources **DITL**. Comme on accède à un dialogue par la ressource **DLOG**, il faut que celle-ci contienne en plus le numéro de la ressource **DITL** associée : il est contenu dans le champ itemsID (figure 2) . Il n'est donc pas du tout obligatoire d'avoir les mêmes numéros pour les deux ressources. Et lorsqu'on déplace un dialogue avec ResEdit, il ne faut pas oublier la partie **DITL**.

Un dialogue est en fait une collection d'items. Chaque item occupe un rectangle dans le dialogue, et a un type précis : bouton, icône, texte, image (voir figure 3). Tous ces paramètres étant éditables avec ResEdit, il est très facile de construire un dialogue ou de modifier son design sans que cela perturbe l'application qui l'utilise.

Au lieu d'utiliser la fonction **NewDialog** dont la déclaration occupe trois lignes, on pourra utiliser la fonction **GetNewDialog** qui va chercher les nombreux paramètres nécessaires à **NewDialog** dans une ressource :

```
FUNCTION GetNewDialog( dialogID : INTEGER ; dStorage : Ptr ;
behind : WindowPtr ) : DialogPtr ;
```

Cet appel charge automatiquement la ressource de type **DLOG** qui a pour numéro dialogID. Avantage : on ne code dans le programme source ni les dimensions, ni le nom ni le style du dialogue, mais seulement un numéro.

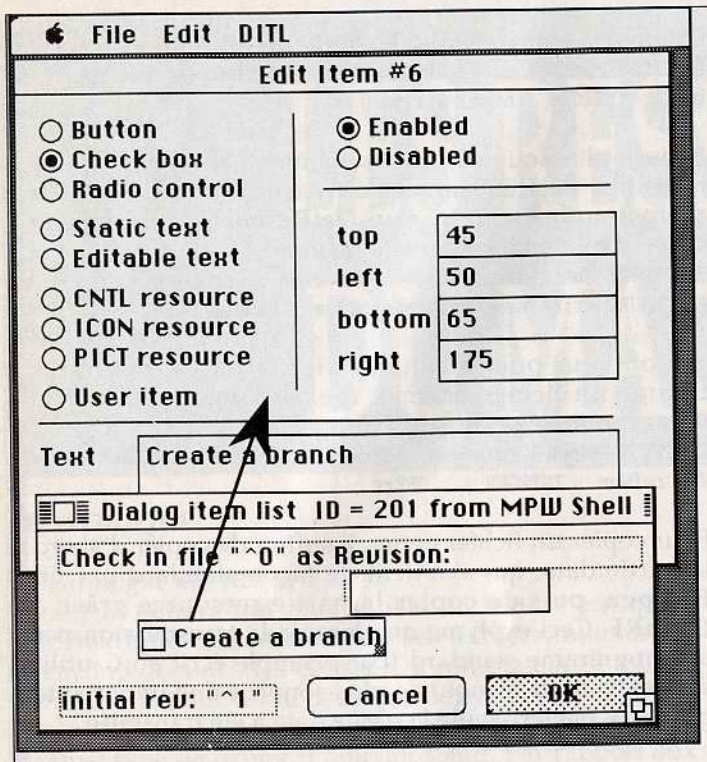


Figure 3 - ressource DITL

Les "Alerts"
Une "alert" permet de prévenir l'utilisateur lorsqu'un événement grave s'est produit, généralement quand le système ne peut plus fonctionner comme prévu. Par sa structure, une "alert" est semblable à un dialogue, sauf que si le système a des ennuis, il se peut que les "alerts" fonctionnent encore lorsque les dialogues sont plantés. Dans ce but, une "alert" est beaucoup moins souple qu'un dialogue : la fenêtre a toujours le même style et elle apparaît toujours au premier plan. Enfin, l'utilisateur doit impérativement traiter l'"alert" avant de faire autre chose. Tout cela fait que les ressources **ALRT** utilisent aussi les ressources **DITL**, étant quasiment identiques aux ressources **DLOG**.

Pour créer une "alert" dont la description se trouve dans la ressource de type **ALRT**, il faut utiliser :

```
FUNCTION Alert(alertID: INTEGER;filterProc: ProPtr):INTEGER ;
```

La simplicité de cet appel encourage les programmeurs à tenir l'utilisateur informé de ce qui se passe.

Les menus
Chaque ressource de type **MENU** contient la description d'un menu déroulant (voir figure 4). Une barre de menus est constituée d'un assemblage de ces menus déroulants. La ressource indique le titre du menu, puis la liste des rubriques, avec pour chacune son style (gras, souligné) et son équivalent clavier éventuel (accessible par Commande). Il est donc très facile de changer ces valeurs, par exemple pour utiliser un équivalent clavier personnel, car le programme se réfère entièrement au contenu des ressources. Il existe aussi une ressource (**MBAR**) donnant une liste de ressources de type **MENU**, permettant ainsi de former une barre des menus.

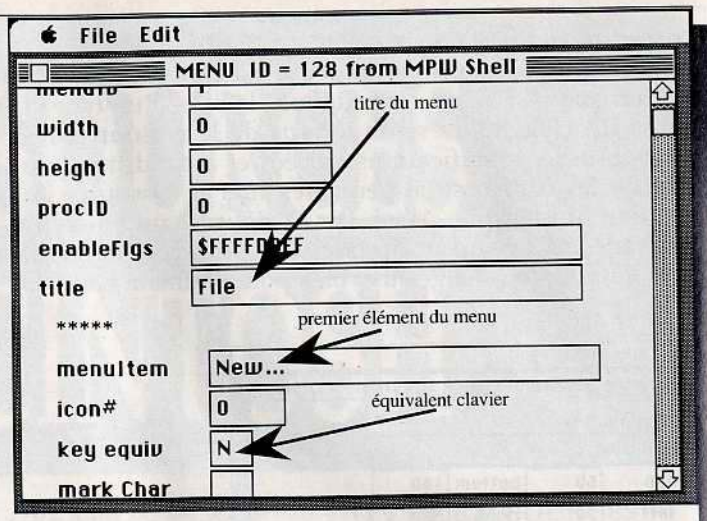


Figure 4 - ressource MENU

Selon que l'on utilise les ressources **MBAR** ou **MENU**, on appellera :

```
FUNCTION GetMenu( resourceID : INTEGER ) : MenuHandle ;
FUNCTION GetNewMBar( menuBarID : INTEGER ) : Handle ;
```

Les fenêtres
Une ressource de type **WIND** contient les dimensions, le style et le nom d'une fenêtre. Le style correspond à la forme du cadre entourant la fenêtre, ainsi qu'à la présence d'éléments comme la "zoom box" ou le titre. La ressource indique également si la fenêtre doit être visible ou masquée lors de sa création, et si elle dispose d'une case de fermeture. ResEdit par défaut montre la fenêtre dans une vue réduite telle qu'elle apparaîtrait sur l'écran actuel, mais il est aussi possible d'afficher la définition de la fenêtre sous forme de texte, afin de modifier au point près les dimensions par exemple.

Comme pour les dialogues, on peut créer une fenêtre en utilisant la fonction **NewWindow** dont la déclaration occupe quatre lignes, mais il est fortement conseillé d'utiliser les ressources grâce à :

```
FUNCTION GetNewWindow( windowID : INTEGER ; wStorage : Ptr ;
behind : WindowPtr ) : WindowPtr ;
```

Les chaînes de caractères
Une ressource de type **STR** contient une seule chaîne de caractères, que le programme peut utiliser comme bon lui semble. Par contre, une ressource de type **STR#** contient une liste de chaînes de caractères, permettant d'économiser de la place lorsque le programme a besoin d'un grand nombre de chaînes. Encore une fois, le fait de placer toutes les chaînes en dehors du code permet une traduction aisée, sans avoir à recompiler en particulier. De plus, cela permet de corriger en une minute des fautes d'orthographe qui seraient désagréables à l'œil.

Pour lire une chaîne de caractères à partir d'une ressource **STR**, on utilisera :

```
FUNCTION GetString( stringID : INTEGER ) : StringHandle ;
```

qui est comme le fait remarquer Inside Macintosh équivalent à :

```
GetResource('STR ',stringID).
```

Dans le cas des ressource **STR#**, il suffira d'utiliser :

```
PROCEDURE GetIndString( VAR theString ; Str255 ; strListID :
INTEGER ; index : INTEGER ) ;
```

Les polices de caractères

Les polices de caractères sont essentiellement stockées dans les ressources de type **FONT**, qui contiennent chacune une seule taille pour une police donnée. Pour une fois, le numéro de ressource a une grande importance, il indique en même temps le nom de la police et sa taille par la formule : ID = (128 * numéro de police) + taille de police. Ainsi, la **FONT** ID=12 correspond à la police Chicago en taille 12.

Le plus souvent, les ressources **FONT** se trouvent dans le fichier System, car elles sont communes à toutes les applications. Cependant, si une application particulière a un besoin spécifique, elle peut contenir les polices correspondantes dans son fichier de ressource personnel.

Le code d'un programme

Un programme compilé se trouve contenu entièrement dans des ressources de type **CODE**, chacune correspondant à un segment. La ressource ayant le numéro 0 est spéciale : elle ne contient pas de code exécutable, mais une structure, la "Jump Table", qui sert à passer d'un segment à un autre. Si le segment n'est pas présent en mémoire, elle appelle le "Resource Manager" pour charger la ressource correspondante. Au lancement d'une application, la ressource CODE ID=0 est chargée, puis la première entrée de la table est exécutée, provoquant le chargement de la ressource CODE ID=1 qui contient le programme principal. Au fur et à mesure que le programme s'exécute, les segments demandés sont chargés en mémoire vive.

Cependant, comme pour toutes les ressources, il est possible de préciser si un segment doit être chargé dans le "application heap" au lancement de l'application. Si c'est le cas, on peut aussi préciser si ce segment (ou plus généralement n'importe quelle ressource) peut être purgé dans le cas ou il y aurait un manque de mémoire ou s'il doit impérativement rester en mémoire vive. Ces fonctions sont assurées par les bits "Locked", "Purgeable", "Protected" et "Preload" dans **Get Info** du menu **File** dans ResEdit. Il est même possible de charger la ressource dans le "system heap" en positionnant le bit correspondant.

Les drivers et accessoires de bureau

Un driver a pour type de ressource **DRVR**, et contient un programme codé en langage machine. Les accessoires de bureau sont considérés sur le Macintosh (et en attendant le Système 7) comme des drivers particuliers. Ils sont donc stockés dans des ressources de type **DRVR**, mais leur ID doit être en principe compris entre 12 et 31. Le plus souvent, tout comme les polices de caractères, ces ressources se trouvent dans le fichier System, mais il peut arriver qu'une application ait besoin de son accessoire de bureau personnel, auquel cas elle pourra l'inclure dans son fichier ressources. Attention, pour déplacer ou installer un accessoire de bureau, il vaut mieux utiliser FontDA Mover, car un accessoire de bureau peut avoir besoin d'autres ressources (des dialogues par exemple), il ne suffit donc pas de copier la ressource **DRVR**.

Pour distinguer les vrais drivers des accessoires de bureaux, ils doivent faire commencer leur nom par un point (.) ou un symbole pourcent (%). Ceci permet d'ajouter facilement dans le menu Pomme la liste des accessoires de bureau, grâce à la procédure :

```
PROCEDURE AddResMenu(theMenu: MenuHandle ; theType: ResType);
```

En effet, cette procédure inclut dans le menu toutes les ressources de type theType, sauf celles dont le nom commence par un point ou un pourcent. L'instruction **`AddResMenu(appleMenu,'DRVR')`** accomplit la bonne action, sans qu'apparaissent dans le menu Pomme les noms de drivers divers.

Et tous les autres...

Il reste encore beaucoup de types de ressources : les icônes (**ICON** et **ICN#**), les images (**PICT**), les curseurs (**CURS**), les sons (**snd**). ResEdit permet de les éditer toutes simplement et il n'y a pas de mystère quant à leur utilisation : c'est toujours le programme qui fournit un numéro et le Resource Manager renvoie l'objet demandé. Enfin, ResEdit est un programme modulaire : il est possible de lui rajouter des sortes de patch (sous forme de ressources bien sûr), qui lui permettent de reconnaître et donc de présenter le contenu des ressources plus agréablement qu'un dump hexadécimal. Par exemple, on trouve ces patch pour les cicn, qui sont des icônes en couleur. Avec cette amélioration constante de ResEdit, l'outil reste à jour et n'est pas dépassé par les nouvelles ressources que les développeurs inventent.

CONCLUSION

Les ressources simplifient la vie du programmeur, car il n'a plus à s'occuper des petits détails que sont les valeurs exactes des paramètres lors de la phase de programmation. Elles simplifient aussi la vie de l'utilisateur car celui-ci peut modifier certains aspects des applications sans aucune connaissance technique (NDLR : ou presque...).

Les ressources posent cependant des problèmes si on les exploite mal : Apple fait remarquer qu'au delà d'un certain nombre de ressources ou d'une certaine taille, le système peut devenir horriblement lent. Il ne faut donc pas que le programmeur soit tenté d'utiliser les ressources comme une base de données bon marché. Il ne faut pas non plus que l'utilisateur modifie n'importe quoi à tord et à travers, sous prétexte que ResEdit est simple d'emploi en apparence. Il faut quand même connaître un minimum le rôle et la signification des ressources, pour pouvoir les manipuler à son gré.

Alain Raynaud

MICRO impression

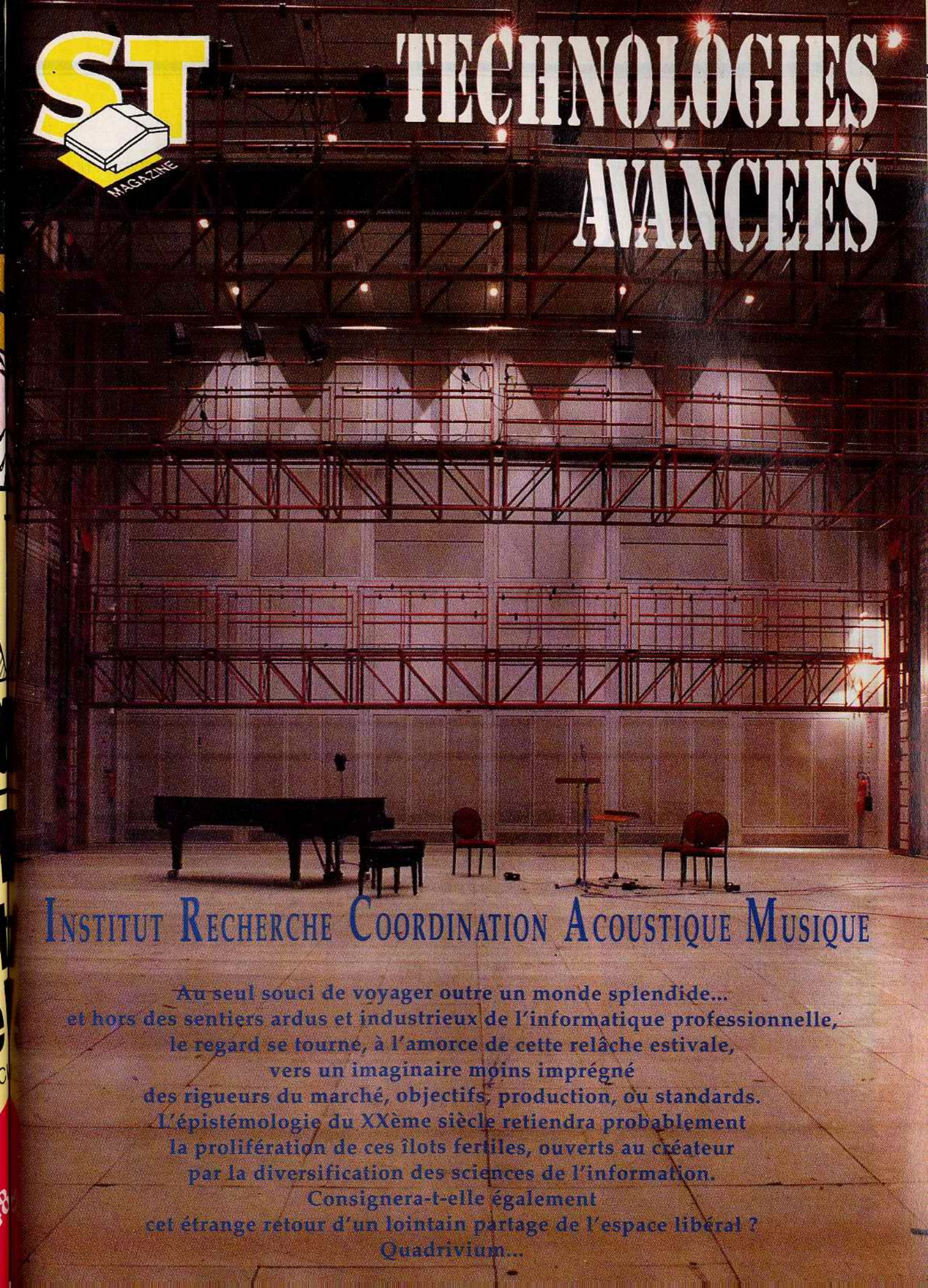
VIDEO PAINT

DESCRIBE

TRUEFORM

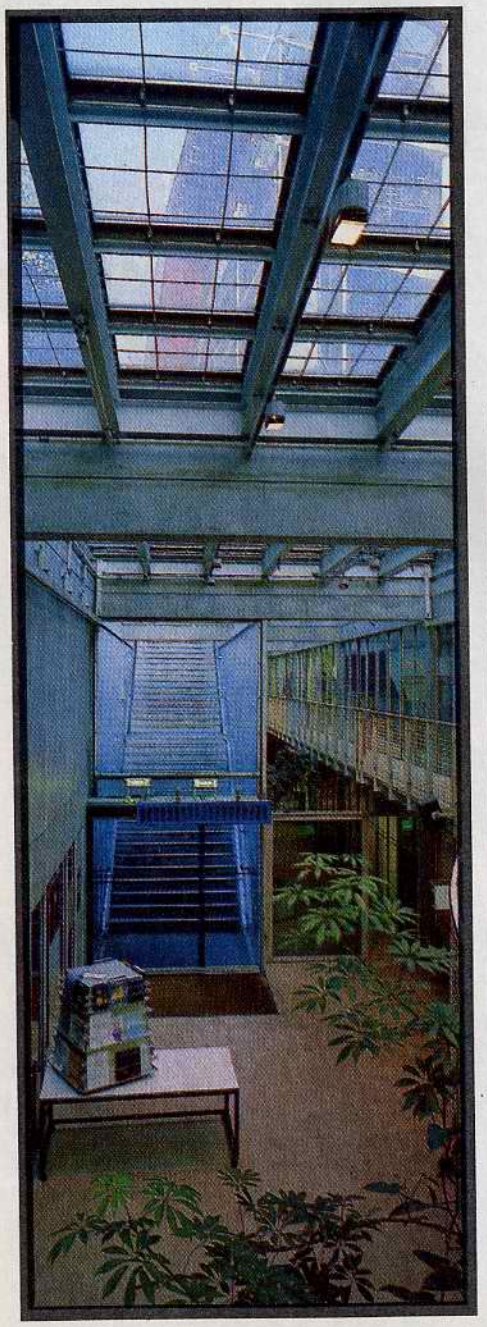
LEXIQUE DE LA TYPOGRAPHIE VIII

M 1201 - 13 - 20,00 F-RD


ST
MAGAZINE
TECHNOLOGIES AVANCEES
INSTITUT RECHERCHE COORDINATION ACOUSTIQUE MUSIQUE
Au seul souci de voyager outre un monde splendide...
et hors des sentiers ardus et industrieux de l'informatique professionnelle,
le regard se tourne, à l'amorce de cette relâche estivale,
vers un imaginaire moins imprégné
des rigueurs du marché, objectifs, production, ou standards.
L'épistémologie du XXème siècle retiendra probablement
la prolifération de ces îlots fertiles, ouverts au créateur
par la diversification des sciences de l'information.
Consignera-t-elle également
cet étrange retour d'un lointain partage de l'espace libéral ?
Quadrivium...

Recherche, Création et Développement

L'IRCAM



CARTE D'IDENTITÉ

L'Institut de Recherche et Coordination Acoustique Musique (situé place Saint-Merri, à Paris) est, aux côtés du Musée National d'Art Moderne, de la Bibliothèque Publique d'Information et du Centre de Création Industrielle, un des éléments du Centre National d'Art et de Culture (CNAC) Georges Pompidou.

Il est régi par un statut d'association d'utilité publique (conforme à la loi de 1901). Son budget de fonctionnement est alimenté (via le CNAC qui fournit locaux administratifs et techniques) par des subventions du Ministère de la Culture, de la Communication et des Grands Travaux (dont le montant pour 1990 s'élève à 30 MF) et par la commercialisation de ses productions et manifestations : concerts, éditions, prestations de services, dons et mécénats, redevances sur les licences des produits industrialisés.

Nous remercions, pour le concours qu'ils nous ont apporté, les responsables de l'IRCAM que nous avons rencontrés lors de la préparation de ce dossier :

- Jean-François ALLOUIS
  Directeur Technique ;
- Laurent BAYLE
  Directeur Artistique ;
- Jean-Baptiste BARRIERE
  Pédagogie ;
- René CAUSSÉ
  Acoustique Instrumentale ;
- Michel FINGERHUT
  Systèmes Informatiques ;
- Andrew GERZSO
  Recherche Musicale ;
- Jean-Pascal JULIEN
  Acoustique des Salles ;
- Eric LINDEMANN
  Développement ;
- Miller PUCKETTE
  Ingénieur ;
- Xavier RODET
  Analyse-Synthèse ;

"Dédicace Spéciale"
à Giuseppe Di GIUGNO,
concepteur du Système 4X.

IRCAM : ACTES INFORMATIQUES

1974
Après quatre ans de travail sur un projet que le président Georges Pompidou lui a proposé de mettre en oeuvre, Pierre Boulez expose les principes directeurs du centre de recherche musicale rattaché au CNAC.

1975
Implantation du centre informatique de l'IRCAM, autour du System 10 de Digital Equipment.

1976
Premier développement scientifique réalisé à l'IRCAM : l'équipe animée par l'ingénieur italien Giuseppe Di Giugno réalise le processeur de sons numériques 4A.

1978
Inauguration de l'Espace de Projection, la salle de concerts interne à l'IRCAM dont les configurations géométriques et acoustiques sont commandées par ordinateur.

1979
Développement, sur un VAX 780 de Digital Equipment, du projet informatique *Chant*, dirigé par Xavier Rodet, pour la synthèse de la voix chantée ; il débouchera sur la mise au point du langage *Formes*, destiné au contrôle des modèles de synthèse, et sera transporté sur un FPS 100.

1981
Création de la version initiale de *Répons*, partition de Pierre Boulez mettant en valeur les outils informatiques créés par l'IRCAM.

1983
Industrialisation du système de traitement de signal numérique 4X par la société Sogitec (pour réaliser un simulateur de vol), aboutissement de la gamme des systèmes inaugurée avec la machine 4A.

1985
Mise au point et présentation, par l'équipe de David Wessel, des premiers logiciels pour micro-ordinateurs (Macintosh) ; organisation des *Ateliers de micro-informatique musicale.*

1986
Symposium *Systèmes personnels et informatique musicale* - concerts présentant des oeuvres réalisées avec les outils micro-informatiques de l'IRCAM (M.-A. Dalbavie, P. Durville, P. Hurel, M. Lindberg).

1987
Esquisse : programme de Composition Assistée par Ordinateur (CAO) réalisé sous la direction de Jean-Baptiste Barrière.

1988
Max/Patcher : logiciel de contrôle des interactions interprète/ordinateur (Miller Puckette).

1989
Développement, sous la direction d'Eric Lindemann, de la *Station d'Informatique Musicale* basée sur l'ordinateur NeXT. Développement d'un environnement de Composition Assistée par Ordinateur (équipe de compositeurs et de techniciens animée par Andrew Gerzso).

PIERRE BOULEZ, DIRECTEUR DE L'IRCAM

Né en 1925, ses activités de compositeur (oeuvres instrumentales, orchestrales et vocales), de chef d'orchestre, de pédagogue et d'organisateur, l'ont placé au premier rang des créateurs de la seconde moitié du siècle.

Présent dans les démarches les plus novatrices de son époque, il fut l'un des tous premiers à saisir l'importance de l'apport technologique dans le champ imaginaire de l'artiste contemporain. Pierre boulez est professeur au Collège de France depuis 1976.

Dominique Jameux a analysé les nombreuses facettes de cette personnalité unique dans sa monographie parue aux éditions Fayard (1984).

RÉPONS

La création, en 1981, de cette oeuvre de Pierre Boulez (version initiale réduite), composée avec l'assistance d'Andrew Gerzso, a marqué une étape importante de l'histoire musicale contemporaine, en constituant le premier chef-d'oeuvre dans lequel les composants contrôlés par l'informatique constituent une partie essentielle du discours musical. Sa version "définitive" (près d'une heure de durée) fut présentée en Avignon, en 1988.

Elle réunit trois types d'intervenants :
- un ensemble de 24 instruments, situé au centre du lieu d'exécution, et entouré par les auditeurs ;
- un groupe de 6 instruments solistes, encadrant le public, dont les sons sont transformés, en temps réel, par le système 4X (filtrages, resynthèse, réinjections, etc.) ;
- un dispositif d'amplification et de spatialisation des sons, transformés ou "naturels", des instruments solistes, contrôlé par ordinateur.

ENTRETIEN AVEC
JEAN-FRANÇOIS ALLOUIS

Les manifestations musicales organisées par l'IRCAM, partie "publique" des activités de l'Institut, sont alimentées par une multiplicité de recherches scientifiques fondamentales et appliquées.
Le Directeur Technique des Activités Scientifiques présente les grands axes de ces travaux, et leurs débouchés dans le secteur industriel.

Dans quel contexte l'équipe scientifique de l'IRCAM a-t-elle entamé ses travaux ?

A l'époque où l'IRCAM a commencé à fonctionner - en 1975 - Pierre Boulez était fasciné par les premières applications d'informatique musicale qu'il avait eu l'occasion de connaître aux Etats-Unis, notamment les premières utilisations d'ordinateur pour synthétiser des sons. En Europe, il n'y avait alors pratiquement rien de fait dans ce domaine, hormis les travaux de certains chercheurs relativement isolés - tels Brown, qui travaillait avec l'INRIA, ou des précurseurs comme Xenakis. Ces travaux se tournaient plutôt en direction de l'utilisation de l'ordinateur dans la composition musicale - générer la partition ou des éléments de partition, ce que nous appelons désormais la composition assistée par ordinateur - mais très peu en direction du son lui-même.

Au départ, l'équipe de l'IRCAM a rassemblé un grand nombre de chercheurs américains, recrutés aux Etats-Unis dans les centres avec lesquels nous entretenons des rapports assez étroits : le CCRMA - le centre de recherche sur l'informatique musicale de Stanford - et l'équipe qui se trouve au Media Lab du M.I.T. Ce fut, en quelque sorte, un transfert de compétence. Les programmes de synthèse musicale en temps différé de l'époque, comme MUSIC V, furent "importés", mais, dans le même temps, une équipe de recherche plus fondamentale, orientée notamment vers l'acoustique, fut créée. Elle comprenait aussi des Français - heureusement ! - comme Jean-Claude Risset, responsable du département informatique, qui venait de travailler aux Bell Labs avec Max Mathews, sur MUSIC V.

Sur quelles bases la démarche scientifique repose-t-elle ?

L'objectif fondamental de l'IRCAM est de mettre les technologies récentes au service de la musique et d'en tirer des outils pour les compositeurs, les musiciens, et les instrumentistes. Mais, par rapport à cette optique très applicative, nous préférons prendre un peu de recul et avoir l'ambition de mener une recherche plus fondamentale. Nous effectuons donc des travaux très applicatifs, mais également des travaux de recherche qui, bien qu'ayant toujours une finalité, n'aboutissent à des applications qu'après plusieurs années d'effort. Ces deux types de travaux s'alimentent constamment, et bénéficient de la présence des compositeurs qui, très concrètement, travaillent sur les oeuvres commandées qui seront exécutées pendant la saison de concerts.

D'autre part, l'IRCAM a été créé à une époque où la technologie d'informatique musicale n'était disponible que dans les centres de recherche. Cepen-

dant, en l'espace de dix ans, elle a été mise sur le marché, et ce changement très important nous a amené à redéfinir notre champ de travail. Il est évident que nous ne faisons pas ce qui est du domaine de l'industrie. Il y a dix ans, nous étions des pionniers, et nous pouvions partir dans toutes les directions. Désormais, le territoire est balisé et nous ne développons que ce qui est de l'ordre de la recherche ou bien ce qui a été laissé de côté pour des raisons économiques, à cause d'une rentabilité insuffisante.

Sur quelles disciplines concentrez-vous vos efforts ?

Les deux disciplines scientifiques avec lesquelles nous avons le plus d'affinités, c'est l'acoustique - à tendance musicale - et l'informatique. Nos travaux d'acoustique s'orientent essentiellement dans trois directions. En acoustique instrumentale, nous étudions le fonctionnement des instruments de l'orchestre : nous en tirons une connaissance plus approfondie de ces systèmes. Ainsi, l'archet numérique sur lequel travaille René Caussé est un système expérimental permettant de mieux comprendre le fonctionnement d'une corde vibrante. Car, pour un phénomène aussi simple - étudié au XIXe siècle par Helmholtz - nous possédons une théorie de base et nous savons à peu près modéliser les phénonèmes de 2e ordre, mais il y a encore des phénonèmes de 3e ordre que nous ne savons pas expliquer !

Au-delà de la pure connaissance scientifique, ces recherches ont un impact sur la lutherie : procédés de conception des instruments, améliorations des instruments existants, etc. Avec le concours de l'université du Maine, nous avons développé pour les luthiers un logiciel de calcul d'impédance acoustique qui tourne sur IBM-PC. Il s'agit d'un logiciel d'assistance à la conception d'instruments qui permet de calculer exactement la position de la perce d'un instrument à vent, position qui donne une assurance sur sa justesse - jusqu'à présent, la détermination de la perce s'effectuait de manière totalement empirique. Nous avons également étudié des systèmes permettant de produire des intervalles microtonaux. La tête de flûte peut, par exemple, être équipée d'un dispositif permettant de produire des quarts de ton et nous pensons adapter ce système au saxophone, à la clarinette, etc.

Notre deuxième champ d'investigation porte sur l'acoustique des salles, et plus généralement, tout ce qui concerne les conditions de reproduction et d'écoute de la musique : le projet est appellé "Salles et sources virtuelles" (1). Dans le cas des salles virtuelles, il s'agit, sans toucher à l'architecture, de changer la personnalité acoustique de la salle, de la corriger avec

La flute 4X et son interface MIDI

VOS ARTICLES NOUS INTERESSENT !!!
TECHNOLOGIES AVANCEES
PRESSIMAGE
19, rue Hégésippe Moreau, 75015 PARIS

des dispositifs électro-acoustiques - en termes techniques on parle de correction active. En situation de concert, les sources sonores sont captées par une prise de son rapprochée, le signal est traité de façon appropriée, puis rediffusé de manière à modifier l'acoustique de la salle.

Quant à la source virtuelle, elle correspond à la reproduction d'une source réelle - un instrument de musique - en prenant en considération toutes ses caractéristiques : non seulement le spectre et le timbre - envisagés sous un aspect unidimensionnel comme c'est le cas dans une prise de son microphonique - mais aussi le rayonnement de l'instrument. On constate, en effet, que la reproduction par haut-parleur, telle qu'on l'utilise actuellement dans la sonorisation d'une salle de concert, n'est pas satisfaisante : elle gêne la plupart des musiciens. Ceux-ci considèrent que, par rapport à l'instrument et à l'acoustique, une dimension a été perdue. Notre hypothèse consiste à chercher une solution du côté du rayonnement, car les caractéristiques de rayonnement d'un haut-parleur sont très spécifiques, et très différentes de celles de la plupart des instruments. En tentant de mieux reproduire ce rayonnement, nous voulons mesurer son impact sur la fidélité de la restitution. Ce sujet de recherche est plutôt d'ordre fondamental, car nous n'avons pas l'espoir de pouvoir réaliser, à court terme, un dispositif miracle.

Le troisième domaine, auquel le groupe dirigé par Steven Mac Adams se consacre, est celui de la perception du son et de la musique. C'est un domaine qui se divise en plusieurs champs de travail. En premier lieu, nous avons le problème du timbre et de la fusion/séparation de sources sonores. En d'autres termes : pourquoi entend-on, à un moment donné, un seul ou plusieurs sons, pourquoi identifie-t-on plusieurs sources virtuelles ? L'analyse porte sur les indices de synchronicité - la cohérence de la modulation des sources sonores - mais sans cependant former d'hypothèse sur le contenu - musical ou non - de la perception sonore.

ESPACE DE PROJECTION

Cette salle unique en son genre est le lieu expérimental des recherches en acoustique architecturale. Elle est également utilisée pour des manifestations publiques (concerts ou enregistrements de l'Ensemble InterContemporain, etc.). Un dispositif contrôlé par un ordinateur Macintosh permet de configurer à l'infini la géométrie et la matière des parois et du plafond (en trois sections indépendantes), formés de modules prismatiques - les périactes - commandés par groupes par trois (171 groupes). Leurs trois faces sont revêtues de matériaux absorbants, réfléchissants et diffusants, et ils sont orientés suivant le type de réaction acoustique souhaité.
Dimensions horizontales : 24 x 15,5 m
Hauteur du plafond : 1,5 à 11,50 m
Capacité d'accueil : 250 à 350 places
Temps de réverbération : de 0,5 à 4,5 secondes.

Plus élevé sur le plan de la cognition, nous menons un travail sur l'interprétation musicale, afin de comprendre les variations d'interprétation d'un même texte, variations entre différents interprètes - ou à différents moments quand il s'agit du même. Nous utilisons pour cela un piano MIDI qui enregistre très précisément les forces d'impact et la durée des appuis sur les touches.

Nous étudions également le phénomène de compréhension musicale : la manière dont un auditeur mémorise et segmente une pièce en plusieurs fragments, et opère un découpage formel. En appliquant des méthodes typiques de la psychologie expérimentale - beaucoup d'analyses statistiques - nous essayons de trouver les éléments pertinents, communs à tous les auditeurs. Cette recherche très fondamentaliste intéresse les compositeurs, car il n'y a pas de règles concernant la manière d'orchestrer les sons synthétiques, et les études sur l'audition peuvent fournir quelques guides, quelques repères, permettant de réaliser un travail d'orchestration à partir de sons synthétiques de n'importe quel type.

Dans le domaine de la synthèse sonore, quels sont vos axes privilégiés ?

Pour produire des sons de synthèse, nous utilisons bien évidemment des produits commerciaux, mais par ailleurs, nous poursuivons nos propres recherches et développements, essentiellement dans deux secteurs : celui de la synthèse par modélisation physique, et celui des synthèses effectuées à partir du modèle vocal.
Jean-Marie Adrien travaille sur les problèmes de modélisation physique en utilisant l'ordinateur pour reproduire, par une simulation complète, un instrument donné. Dans le cas du violon, le processus consiste d'abord à déterminer une description géométrique et mécanique du violon, ensuite, à rentrer ce modèle formalisé dans la machine, et enfin, à faire calculer par la machine, en appliquant les équations générales de la mécanique, la manière dont le violon fictif va vibrer. Dans cette simulation, les limitations pratiques proviennent de la masse considérable de calculs qu'il faut effectuer et, il y a dix ans, ce processus n'était pas envisageable. Mais la puissance de calcul des machines ayant évolué exponentiellement, ces méthodes deviennent applicables. Elles ont l'avantage de permettre une reproduction fidèle du son instrumental, notamment au niveau des transitoires et des phénonèmes d'attaque, une des parties spécifiques du timbre.

Il faut aussi souligner que, dans les méthodes de synthèse traditionnelles - celles qui sont mises en oeuvre dans les synthétiseurs commercialisés - le signal produit n'a aucune dimension spatiale. La méthode de production étant totalement arbitraire, même si le son est reproduit en stéréophonie, la distribution dans l'espace de ce signal ne correspond pas à la manière dont le son devrait être spatialisé. A l'inverse, en réalisant la modélisation physique d'un instrument, la vibration de chaque point de sa surface rayonnante peut être calculée. Ainsi, dans le cas du violon, c'est essentiellement la caisse qui rayonne. En se plaçant à l'extérieur, dans la position d'un auditeur ou d'un microphone, la contribution parvenant de chacun des points est très exactement quantifiable - éventuellement, la réverbération de la salle d'écoute peut être prise en compte. Il est donc possible

d'obtenir des signaux de synthèse réellement spatiaux - en fait, ayant un rapport avec la position dans l'espace de l'instrument ; on peut même simuler une prise de son stéréophonique. La modélisation physique a l'avantage de donner accès à la dimension spatiale du son, de manière naturelle et rationnelle, et non pas arbitraire, comme c'est le cas avec les autres méthodes de synthèse.

D'un autre côté, l'IRCAM a la chance d'avoir une équipe de travail sur la synthèse de la voix, qui est une des meilleures au niveau mondial. Le modèle vocal est un modèle général très intéressant (1), car la voix est beaucoup plus complexe que la plupart des instruments de musique. Nous avons commencé par synthétiser la voix chantée, mais, avec la même technique, il est tout à fait possible de réaliser des synthèses instrumentales très intéressantes.

La station musicale de l'IRCAM permettra-t-elle d'aller plus loin dans ces travaux ?

Nous avons besoin d'un système qui, a notre avis, n'existe pas actuellement sur le marché, notamment pour faire de la recherche sur la synthèse et travailler avec des techniques d'informatique musicale. Ce système doit avoir pas mal de puissance de calcul et de capacité temps réel - puisque c'est ce que l'on demande à un système qui doit faire de la synthèse sonore dans le contexte d'un concert. Il doit aussi être largement ouvert, car l'IRCAM est un centre expérimental dont l'ambition est d'offrir ce qui n'est pas disponible ailleurs. Une telle position nous conduit à faire beaucoup de prototypage, afin d'apporter une réponse aux idées des compositeurs et, pour permettre le développement rapide d'un court logiciel spécifique, la qualité de l'environnement de programmation est essentielle. Enfin, par rapport au milieu musical, le coût du système doit être raisonnable : il faut que nous puissions en acquérir plusieurs, et que d'autres centres - ou même des musiciens individuels prêts à faire l'investissement - puissent s'en équiper. En fait, il ne doit pas coûter plus cher qu'un piano de concert.

Voilà la raison d'être de ce projet, qui cherche à intégrer les meilleures techniques afin de créer une plateforme de travail sur laquelle nous allons continuer à développer des logiciels de CAO, de synthèse, etc.

Dans quelles directions les recherches vont-elles se poursuivre ?

Il faut continuer à appliquer les nouvelles idées, même celles qui sont encore au stade des balbutiements. Nous allons reprendre l'étude d'un système artificiel capable d'analyser un environnement sonore - tout comme on écoute un interprète - et d'en retirer les informations qui lui permettent de réagir. Si nous avons actuellement de bons systèmes de synthèse sonore, nous avons des difficultés à contrôler ceux-ci de façon intelligente et à les faire jouer : ils se trouvent donc isolés dans un univers artificiel, incapables de communiquer naturellement avec les musiciens vivants. C'est une situation analogue à celle de la parole : on sait réaliser la synthèse, mais pas la reconnaissance vocale - tout au moins à un niveau complexe. Or, les applications de la reconnaissance de la musique par une machine sont multiples. Il y a notamment la transcription automatique des partitions : l'enjeu économique en est certainement important, mais nous sommes plutôt attirés par l'idée d'accompagnement d'un interprète par une machine.

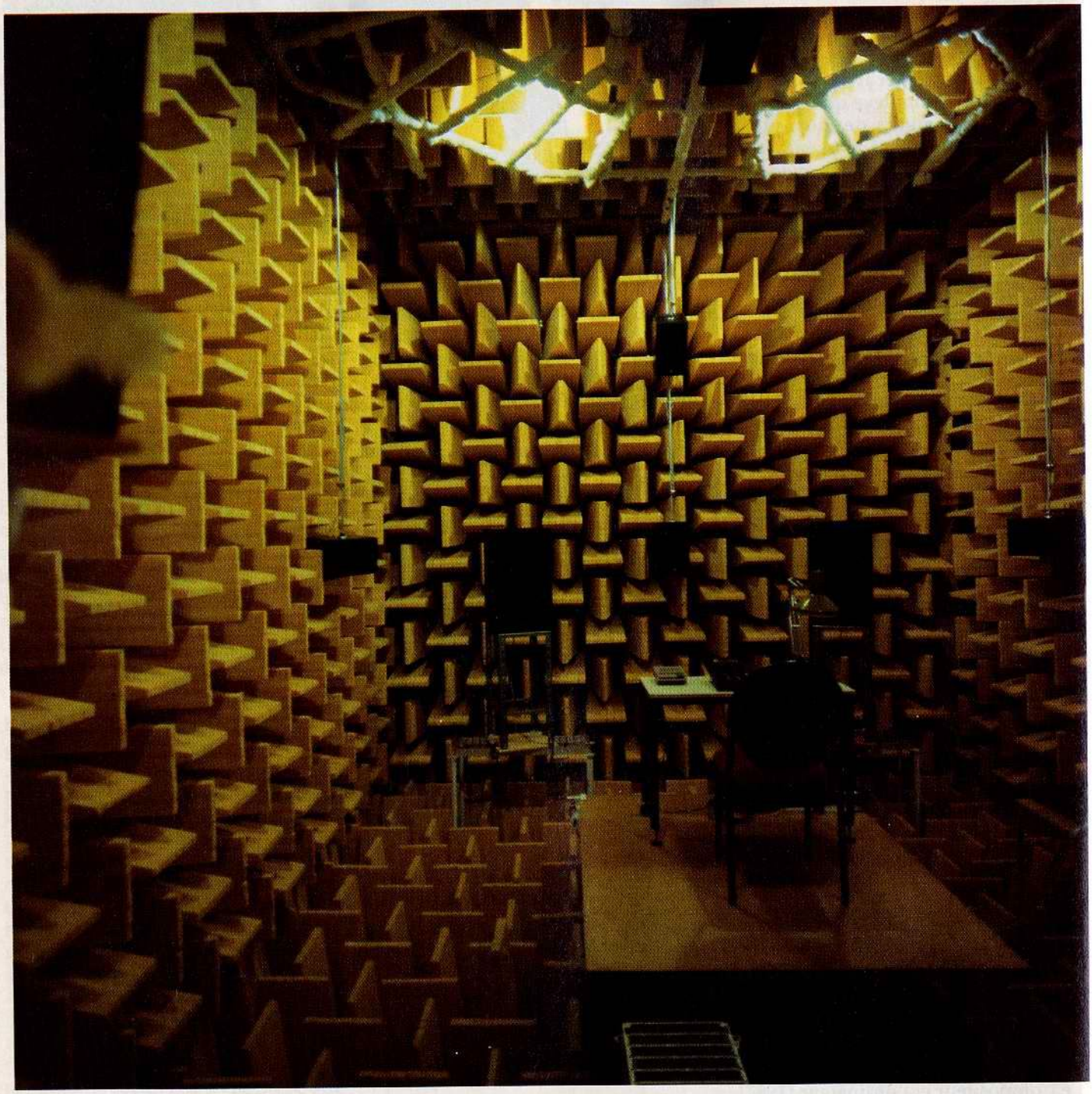
Nous avons déjà avancé dans cette voie : des instruments ont été équipés de capteurs analysant la position des clés - c'est le principe de la flûte qui envoie des codes MIDI. Un programme "connaissant" la partition que doit jouer l'instrumentiste, peut, à partir des informations transmises par les capteurs, se repérer dans cette partition, et en conséquence, jouer sa propre partie. Toutefois, étant donné que les capteurs ne sont pas toujours utilisables, nous voulons aller plus loin, afin de travailler directement à partir du son émis.

De façon générale, nous axons nos recherches sur la modélisation de

RÉFÉRENCES (Ouvrages édités par le CNAC)
Rapports de Recherche :
The Computer as a sound processor - n• 25
Chant, de la synthèse de la voix chantée ... - n• 35
Formes, composition et ordonnancement de processus - n• 36
La Station de travail 4X - n• 39
La Composition assistée par ordinateur - n• 41
Le Compositeur et l'ordinateur (1981)
Actes de la Conférence Internationale d'Informatique Musicale (1984)

tout ce qui se rapporte au contrôle et à l'interprétation, de manière à ce que, à un niveau élémentaire, la machine puisse intervenir automatiquement. Les variations d'interprétation, les problèmes d'articulation entre les sons, la manière de les jouer, sont des problèmes que nous rencontrons dans la synthèse. Mais il nous reste à trouver la façon dont les sons s'articulent entre eux - tout comme ils le sont dans un jeu instrumental normal - puis à modéliser les règles de jeu, les règles pour passer d'un son à un autre, etc. Enfin, il faut transférer cette connaissance dans la machine, ceci dans le but de réaliser une simulation du jeu instrumental.

Dans le domaine de l'acoustique des salles, un de nos buts est d'arriver à des dispositifs utilisables en concert. Mais nous avons également un projet qui s'adresse aux architectes concepteurs de salles. Avec le package de logiciels que nous développons, un architecte sera à même de projeter les caractéristiques acoustiques de la salle qu'il doit dessiner et, en fonction des défauts qu'il peut constater, faire les modifications architecturales qui conviennent. Ce qui est important, c'est que l'archi-



La chambre anéchoïque de l'IRCAM

tecte pourra utiliser ce logiciel directement, sans faire appel à l'expertise d'un constructeur acoustique - de la même manière qu'il utilise un programme de C.A.O.

Comment les équipes de chercheurs sont-elles constituées ?

Les travaux de recherche sont souvent réalisés par un chercheur permanent et deux ou trois stagiaires - des étudiants réalisant une thèse ou un DEA. Il y a également de nombreuses collaborations : dans le domaine instrumental nous travaillons beaucoup avec l'Université du Maine, avec des luthiers, et, bien sûr, en liaison avec l'Ensemble Intercontemporain - ses instrumentistes, qui ont un rapport privilégié avec l'IRCAM, apportent des idées et nous servent un peu de "cobayes".

L'IRCAM entretient-il des liens étroits avec les industriels et la communauté scientifique ?

Nous collaborons avec des universités françaises et étrangères - américaines pour la plupart. Dans le domaine de la recherche fondamentale, des étudiants effectuent leurs stages de thèse à l'IRCAM sur des sujets que nous définissons. Parmi les universités avec lesquelles nous entretenons des relations quasi-permanentes, on compte notamment l'université du Maine - DEA d'acoustique appliquée - ; le LAFORIA de Paris VI - laboratoire informatique travaillant dans le domaine de l'intelligence artificielle - ; le LIMSI d'Orsay - laboratoire d'informatique appliquée s'intéressant à la communication avec les machines. Il y a également des contacts plus ponctuels : par exemple, un des membres de l'équipe travaillant sur la station musicale prépare, dans le cadre du laboratoire d'informatique du parallélisme de Lyon, une thèse sur les problèmes de compilation des programmes pour les processeurs que nous mettons en oeuvre.

Sur le plan international, nous entretenons des rapports privilégiés avec Standford, San Diego, le M.I.T., et en Europe avec l'université de Stockolm.

Au niveau industriel, la collaboration est plus épisodique, et n'intervient que lorsque nous sommes certains de déboucher sur un produit ayant un intérêt commercial, ou lorsque nous avons trouvé une solution susceptible d'avoir un impact industriel. Dans le passé, nous avons commercialisé - avec la firme Sogitec - le système 4X, mais surtout pour des applications extra-musicales - malheureusement le produit était trop coûteux pour pouvoir trouver un créneau sur le marché musical. Il y a deux ans déjà, notre collaboration avec Act Informatique a débouché sur la réalisation commerciale de MidiLisp - une version de l'environnement Le_lisp destinée à la réalisation de programmes musicaux. Dans un proche futur, un industriel américain (2) va commercialiser la station musicale dont nous achevons la mise au point, et avec d'autres compagnies nous allons mettre sur le marché plusieurs logiciels développés à l'IRCAM - logiciels MIDI notamment. Ces logiciels sont orientés vers la recherche et la pratique de la musique contemporaine, mais leur potentialité permet de les adapter aux besoins du marché.

De quelle manière la collaboration avec les industriels se concrétise-t-elle ?

Dans la majorité des cas, nous sommes amenés à chercher des partenaires. Cependant, dans le secteur de l'acoustique, les professionnels de la lutherie sont très attentifs à nos travaux. Au départ, l'assistance informatique ne faisaient pas partie de leur méthodes de travail, mais à partir du moment où nous avons organisé des séminaires pour expliquer ce que nos logiciels pouvaient leur apporter, le mouvement s'est déclenché : une dizaine de luthiers en France les utilisent maintenant et sont passionnés.

Il faut souligner aussi que, dans le cadre du mécénat, nous avons bénéficié de l'aide de grands industriels. Sur le plan technologique, Apple, à l'époque du lancement du Macinstosh, nous a fourni un certain nombre de machines. Aujourd'hui, la très importante donation de Digital Equipment vient de permettre le renouvellement d'une partie de notre matériel informatique. Ce soutien s'adresse directement au côté recherche de l'IRCAM, mais beaucoup d'autres compagnies nous prêtent leur concours dans le domaine artistique - là où l'image est plus visible - en aidant à monter des concerts ou des opérations de production.

Il existe aussi une autre forme de mécénat, moins spectaculaire, mais très intéressante. Ainsi, il y a un an, nous avons monté une opération de mécénat technologique avec l'EDF : l'équipe de recherche en acoustique de cet organisme nous a fait bénéficier de son expertise dans les programmes de simulation. Ensemble, nous avons pu développer une opération de contrôle acoustique de notre Espace de Projection.

Comment les activités de l'IRCAM sont-elles diffusées au près des chercheurs et du public ?

Dans le domaine scientifique, la diffusion des résultats adopte une forme classique : publication dans les grandes revues scientifiques concernées, conférences, etc. - de ce point de vue, nous fonctionnons comme le CNRS. Sur le plan technologique, nous avons publié des rapports de recherche, et nous allons continuer dans cette voie - en collaboration avec la Direction de la Musique qui veut publier des rapports sur l'ensemble des recherches en France. Enfin, chaque année, nous réalisons un rapport d'activités présentant l'ensemble du travail scientifique effectué.

Propos recueillis par
Daniel Fournier

(1) Voir, dans ce même numéro, l'article se rapportant à ce sujet.

(2) La société Ariel Corp., située dans le New Jersey

ACOUSTIQUE ARCHITECTURALE

Versant technologique de la recherche musicale, la connaissance des phénomènes acoutisques s'inscrit dans les territoires d'investigation privilégiés de l'IRCAM. Pour pouvoir se déployer sur deux axes convergents, source/propagation, des outils informatiques spécifiques ont été développés : systèmes d'analyse, logiciels de contrôle, interfaces de configuration ...

Les deux axes considérés s'illustrent par des recherches menées concurremment dans les domaines de l'acoustique instrumentale (les éléments producteurs de phénomènes vibratoires), et de l'acoustique architecturale (l'environnement de propagation des vibrations sonores). C'est le second qui retiendra notre attention, en raison de l'importance des développements informatiques qui le sous-tendent.

C'est, en effet, dans cette direction que les développements logiciels les plus importants ont été accomplis - et actuellement poursuivis - dans le cadre d'une recherche portant sur la maîtrise de "l'effet de salle". S'étendant au-delà des limites propres de l'étude des composants acoustiques des lieux de diffusion du son (salles de spectacles, amphithéâtres, etc.), elle s'étire aussi bien en amont, afin d'explorer les composantes psychologiques de l'appréciation des volumes acoustiques (validation perceptive de la caractérisation objective), qu'en aval, en cernant ses terrains d'application (commande interactive de l'Espace de Projection de l'IRCAM). Ces études mettent à contribution les techniques d'analyse et de traitement du signal tout autant que le formalisme mathématique de l'acoustique physique.

L'équipe formée à l'IRCAM s'est successivement penchée sur les problèmes de :

- métrologie acoustique, et modélisation paramétrique ;
- perception auditive ;
- techniques informatiques appliquées à l'acoustique prévisionnelle.

Ce dernier sujet est d'ailleurs en pleine progression et constitue l'essentiel des travaux en cours, avec, pour perspective d'application, la production artificielle de champs sonores.

Sources virtuelles

Cette étude aborde le problème de la restitution des phénomènes perceptifs liés à la diffusion d'une source sonore dans une salle. En relevant les paramètres objectifs déterminés par les études antérieures, le contrôle s'exerce sur deux facteurs primordiaux :

- l'onde directe propagée (neutralité de la fonction de transfert du transducteur diffusant l'onde directe) ;
- le champ réverbéré (identité de l'indice de directivité du dispositif de diffusion).

Les systèmes expérimentaux élaborés (jeux de haut-parleurs, baffle dodécaédrique) sont dotés d'éléments de traitement du signal (filtrage, matriçage, etc.) afin de constituer un outil de simulation puissant.

Aide à la conception architecturale

Sur le plan informatique, c'est dans cette direction que les efforts les plus importants sont effectués, afin de réaliser des outils logiciels utilisables par les architectes dans leur travail de conception des lieux d'écoute. L'expertise acquise par le laboratoire a guidé le choix des méthodes informatiques utilisées. Elle a permis de dégager deux grandes options stratégiques :

- simulation de la propagation acoustique ;
- recherche de procédures d'inversion, en vue de modifications apportées à un projet architectural.

Les développements se sont donc concentrés sur la mise au point de trois types d'outils, considérant :

- la mesure des paramètres acoustiques ;
- le diagnostic acoustique ;
- l'aide à la conception des lieux d'écoute.

L'Espace de Projection de l'IRCAM

Mesures in situ
Il a été nécessaire de corriger et compléter les informations fournies par les fabricants de matériaux de construction, dont les valeurs (approximatives, voire fausses !) sont utilisées dans les programmes de simulation. Une base des mesures des critères acoustiques de l'Espace de Projection, servant de lieu expérimental, a tout d'abord été constituée. Une confrontation de ces données avec une étude théorique et expérimentale a permis d'affiner les valeurs utilisables.

Critères du diagnostic acoustique
L'achèvement, en 1989, des travaux de recherche sur la perception auditive de la qualité acoustique des salles a fourni des résultats permettant d'établir des relations structurées entre critères acoustiques mesurables et facteurs perceptifs objectifs. L'interfaçage de ces nouveaux critères d'évaluation a été intégré au projet.

Conception des lieux d'écoute
Après examen systématique des applicatifs disponibles, un environnement logiciel complet a été construit, afin de servir de base aux travaux de recherche sur les procédures d'inversion.

Environnement logiciel
Le développement des interfaces homme/machine des outils utilisés a été confié à la société APIA, qui a adapté ses programmes existants. Tournant sur PC et Windows 2, ils servent d'interface d'E/S, et permettent de transmettre les données expérimentales aux programmes de simulation, sous forme de fichiers ASCII. La normalisation de l'ensemble de ces programmes, tournant primitivement dans des environnements hétérogènes (PC, Vax 780, Fortran, Pascal, etc.), a été opérée, dans le souci d'une meilleure portabilité. Le langage C a ainsi été choisi comme support de développement.

Le système ainsi élaboré est architecturé de manière à rendre possible l'évolution des éléments de la simulation en modularisant les différentes hypothèses acoustiques et fonctions. Quatre modules complémentaires sont mis en oeuvre :
- "son direct/diffraction réflexion/cônes" se base sur l'hypothèse acoustique de réflexion spéculaire et décrit, en fonction du temps, la puissance captée par la surface et par les récepteurs distribués dans l'enceinte du lieu d'écoute ;
- "diffus direct" récupère les données captées, et commande la propagation de la puissance acoustique non-réfléchie spéculairement ;
- "diffus indirect" fournit une description sous forme matricielle du champ réverbéré, indépendamment de l'excitation de la salle ;
- "intégration" opère la synthèse des informations calculées par les trois programmes précédents, et construit les courbes énergie/temps des récepteurs utilisés pour le diagnostic acoustique.

Des logiciels existants ont également été adaptés à l'expérimentation de l'Espace de Projection, et leur évaluation a été effectuée dans un cadre d'étude concret : la salle de concert de la Cité de la Musique (Paris-La Vilette).

Commande perceptive de l'Espace de Projection
Depuis sa création, l'Espace de Projection est doté d'un logiciel de configuration spatiale de ses parois, agissant sur le positionnement des périactes, ceci afin de pouvoir modifier la qualité acoustique de la salle. Il reste cependant délicat à utiliser pour des non-spécialistes, car il présuppose une connaissance approfondie des relations entre le positionnement des périactes et la réponse acoustique de la salle. Un programme est donc développé (en collaboration avec le Laboratoire acoustique de l'EDF) afin de permettre au "profane" (acousticien, régisseur, interprète, compositeur) de configurer aisément cet espace acoustique.

Interface facteurs perceptifs/critères objectifs
L'interface du logiciel rend possible l'utilisation de critères objectifs (pertinents pour les acousticiens), qui décrivent la qualité acoustique de la salle, évaluée par l'utilisateur au travers de facteurs perceptifs. En retour, le contrôle des données spatiales de la salle permet d'établir une correspondance avec cette évaluation auditive. L'utilisateur dispose d'un tableau de bord décomposé en deux domaines d'interaction, présentant la liste des critères objectifs et des facteurs perceptifs, chacun assorti des paramètres qui lui conviennent. La gestion des relations partielles ou univoques entre facteurs perceptifs et critères objectifs est opérée par des boîtes contenant des tables de conversion.

Processus d'inversion
La modification d'un paramètre d'un des domaines provoque les conversions nécessaires dans le domaine complémentaire, et la propagation à tous les autres paramètres (suivant un ordre prédéfini), des contraintes objectives qui peuvent s'attacher à chaque facteur. Une fois la validation du choix de l'utilisateur effectuée, les données calculées sont transmises au programme d'automatisation de la configuration.

Ce processus pose des problèmes délicats à résoudre. La détermination d'une fonctionnelle, représentant globalement le comportement acoustique de l'Espace de Projection, a été rendu possible en construisant une base de fonctions, répertoriant les données spatiales pour lesquelles la valeur de chaque critère peut être décrite sous forme de combinaison linéaire. La matrice de passage qui découle de cette formalisation utilise la base contenant les mesures effectuées dans l'Espace de Projection. Pour un ensemble rassemblant configurations spatiales et valeurs des critères acoustiques correspondants, un programme effectue une estimation de la meilleure description des critères objectifs. Lorsqu'une approximation convenable a été trouvée, l'exploitation de la matrice inverse, calculée par inversion de la matrice de passage suivant une procédure itérative optimisée, permet d'aboutir à la configuration spatiale recherchée.

Il est clair que l'ensemble de ces recherches mettent en oeuvre des ressources expérimentales sophistiquées, aussi bien sur le plan formel que matériel. L'importance des travaux d'acoustique réalisés à l'IRCAM, dont l'Espace de Projection constitue le lieu privilégié d'expérimentation, peut d'ailleurs être soulignée par l'intérêt que lui portent plusieurs sociétés industrielles, parmi lesquelles on note la participation active du Laboratoire d'acoustique de l'EDF, ou des Laboratoires Bell d'AT&T.

Daniel Fournier

LA STATION MUSICALE IRCAM

Point d'aboutissement d'un développement de près de deux années, la station musicale IRCAM représentera, à la fin de l'année 90, l'outil audionumérique le plus sophistiqué présent sur le marché micro-informatique. Deux champs d'application complémentaires ont guidé sa conception : la composition musicale assistée par ordinateur et le traitement du signal audionumérique.

En lançant le projet de station musicale, l'IRCAM a souhaité répondre, non seulement à des besoins internes, mais également aux exigences du marché de la micro-informatique musicale, lequel doit faire face à des impératifs de puissance et de complexité chaque jour plus pressants. La sophistication des matériels qui équipent les studios audionumériques nécessite en effet des organes de contrôle dotés de capacités se situant, de plus en plus fréquemment, au-delà de celles qui sont offertes par les systèmes utilisés actuellement.

ARCHITECTURE

Les impératifs auxquels l'architecture de la station musicale de l'IRCAM doit faire face sont multiples :

- puissance de calcul élevée, afin de procéder au traitement des événements et des signaux d'un système d'exécution musical complexe ;
- système d'exploitation temps réel, afin de correspondre à l'environnement de travail habituel du musicien, et multiprocesseur, afin de gérer la multiplicité des cartes, nécessaire pour obtenir la puissance de traitement recherchée ;
- périphériques de contrôle supportant des interfaces interactives pour la visualisation des programmes de commande, l'enregistrement et l'édition de leurs exécutions.

"Le calcul en temps réel pour la musique (traitement de signal et gestion d'événements en temps réel dans le contexte d'un concert, etc.) constitue notre principal centre d'intérêt. Les systèmes temps réel relativement puissants sont encore assez difficiles à programmer, plus difficiles que les ordinateurs standard. Ainsi, le système dont nous disposons actuellement est réparti sur trois niveaux : une partie machine dédiée au traitement de signal (microcodée), contrôlée par un ordinateur temps réel (à base de microprocesseur MC68000, 20, 30, ou Intel i386, etc.) chargé de la gestion des événements, l'ensemble étant dirigé par un autre ordinateur (système Unix, Macintosh, Atari, etc.).

"Cette configuration est assez complexe, les environnements de développement différents - celui du traitement de signal étant assez difficile à maîtriser. Ordinairement, la recherche ne se fait pas dans ce genre de contexte, mais sur des machines plus standard (VAX, station UNIX ou encore Macintosh), et quand on trouve un algorithme satisfaisant, on tente de l'adapter à ces systèmes plus complexes : à l'IRCAM, c'est le système 4X, l'architecture dont je viens de parler.

"Pour la nouvelle génération de machines, nous avons donc voulu simplifier la structure, afin que les chercheurs puissent travailler directement sur une machine temps réel puissante, et construire un environnement bien adapté à la recherche d'algorithmes temps réel, permettant de réaliser facilement des prototypes."

Eric Lindemann,
Responsable des développements matériels et logiciels à l'IRCAM

Multiprocesseur

Le principe retenu a été celui de la carte d'extension insérable dans le fond de panier d'une machine standard. Chaque carte est déjà un système multiprocesseur en soi, en juxtaposant :

- 2 microprocesseurs Intel i860, pour le traitement des signaux et la gestion des événements temps réel ;
- 1 microprocesseur Motorola DSP56001, pour la gestion des E/S de la carte (MIDI et audionumérique).

L'échange de données entre les microprocesseurs est assuré par un commutateur à barres croisées. La carte comporte également une RAM locale de 8 Mo (extensible jusqu'à 32 Mo). Une carte annexe est prévue pour fournir 4 canaux d'E/S audio-analogiques.

Le système atteint un second niveau multiprocesseur par la multiplica-

"Nous avons voulu construire un système à base de carte insérée dans une machine hôte déjà présente sur le marché. Nous ne souhaitions pas construire de système complet indépendant, à cause des problèmes de commercialisation : il est beaucoup plus facile de vendre une simple carte destinée à un système existant.

"Cette approche a posé des contraintes sur le choix du système hôte. Nous avons besoin d'un système possédant une bande passante de disque relativement élevée, permettant à l'hôte de tourner pendant les transferts DMA sur disque. Il faut pouvoir enregistrer sur disque (ou reproduire) jusqu'à quatre canaux audionumériques en temps réel. Le bus système doit être performant et le format de la carte suffisamment grand pour pouvoir réaliser une carte complexe.

"NeXt possède tout cela : un bon disque DMA (un disque SCSI typique), ce qui n'est pas le cas du Mac ; les cartes des PC sont relativement petites, alors que celles du NeXT mesure 1 pied carré ; le bus est un NuBus amélioré par un mode "rafale" assez performant - en théorie, le taux de transfert atteint 100 Mo/s, mais en réalité à peu près le tiers, ce qui reste très supérieur à tous les PC. C'est donc le seul système qui possède ces qualités dans une configuration relativement bon marché, meilleure, en tous cas que les serveurs Unix.

"J'ajouterais qu'un gros effort a été accompli sur l'interface, pour réaliser, sur un système Unix, une boîte à outils comparable à celle à du Mac. A mon avis, il s'agit d'une interface utilisateur plus avancée que celles des standards actuels du marché, plus pensée et complète qu'X Window-Motif. Malgré tout, le fait que ce ne soit pas un standard nous pose quand même pas mal de problèmes."

Eric Lindemann

tion des cartes à l'intérieur de chaque machine, et un troisième par l'interconnexion des machines entre elles, au moyen de liaisons parallèles à haut débit. La puissance globale du système est donc largement adaptable aux situations les plus complexes.

"En plus des deux processeurs i860, nous avons placé sur la carte un processeur Motorola DSP56000, mais sans fournir cependant d'accès à sa programmation. Il sert surtout de processeur d'E/S.

"Il a été choisi car il possède de bons ports sériels, qui peuvent, par programmation, fonctionner à des vitesses différentes. La carte dispose ainsi de 4 canaux d'E/S audionumériques - ce qui est raisonnable, même si l'on veut avoir toujours plus - et de ports série plus lents pour les E/S MIDI. Et comme il reste un peu de puissance, nous avons intégré le logiciel de contrôle DMA pour gérer les communications entre les i860 répartis sur différentes cartes."

Eric Lindemann

Machine hôte

L'interfaçage des cartes de traitement de signal avec l'utilisateur du système s'articule autour d'un système hôte. Celui-ci doit être capable de performances élevées dans plusieurs domaines à la fois :

- transfert de données, pour les E/S (bus système à haut débit) ;
- graphisme (interface homme/machine), pour l'interactivité ;
- environnement de développement, pour la programmation d'applications.

C'est l'ordinateur NeXT qui a été finalement choisi en regard de ses qualités évidentes sur chacun de ces points :

- 4 emplacements sur le fond de panier (dont un est occupé par la carte-mère);
- bus système de technologie CMOS, à la norme IEEE 1196 (NuBus), mais dont la vitesse de transfert a été doublée (100 Mo/s) par un mode rafale (*burst mode*) ;
- système de fenêtrage très performant (Display Postscript), développé, dans l'esprit de X Window System, à partir du langage de description de page d'Adobe System ;
- système d'exploitation compatible avec UNIX (basé sur le noyau Mach), offrant une vaste gamme d'outils de développement (langages de programmation par objets, etc.).

EXÉCUTIF TEMPS RÉEL

Le caractère des opérations effectuées en traitement de signal, soumises aux impératifs du temps réel, a induit le développement d'un noyau temps réel spécifique, nommé IMRTX, adapté au microprocesseur i860. Il constitue une couche supérieure au noyau de base CPOS (*Coprocessor Operating System*) qui fournit :

- les primitives de communication entre le système de traitement multiprocesseur (i860) et l'ordinateur hôte (NeXT) ;
- les fonctions de gestion de la mémoire (protection, allocation, etc.) ;
- les primitives de mise au point (*debugging*).

Cet exécutif s'interface avec le système d'exploitation du NeXT par un mécanisme d'échange de messages. Ainsi peut s'effectuer :

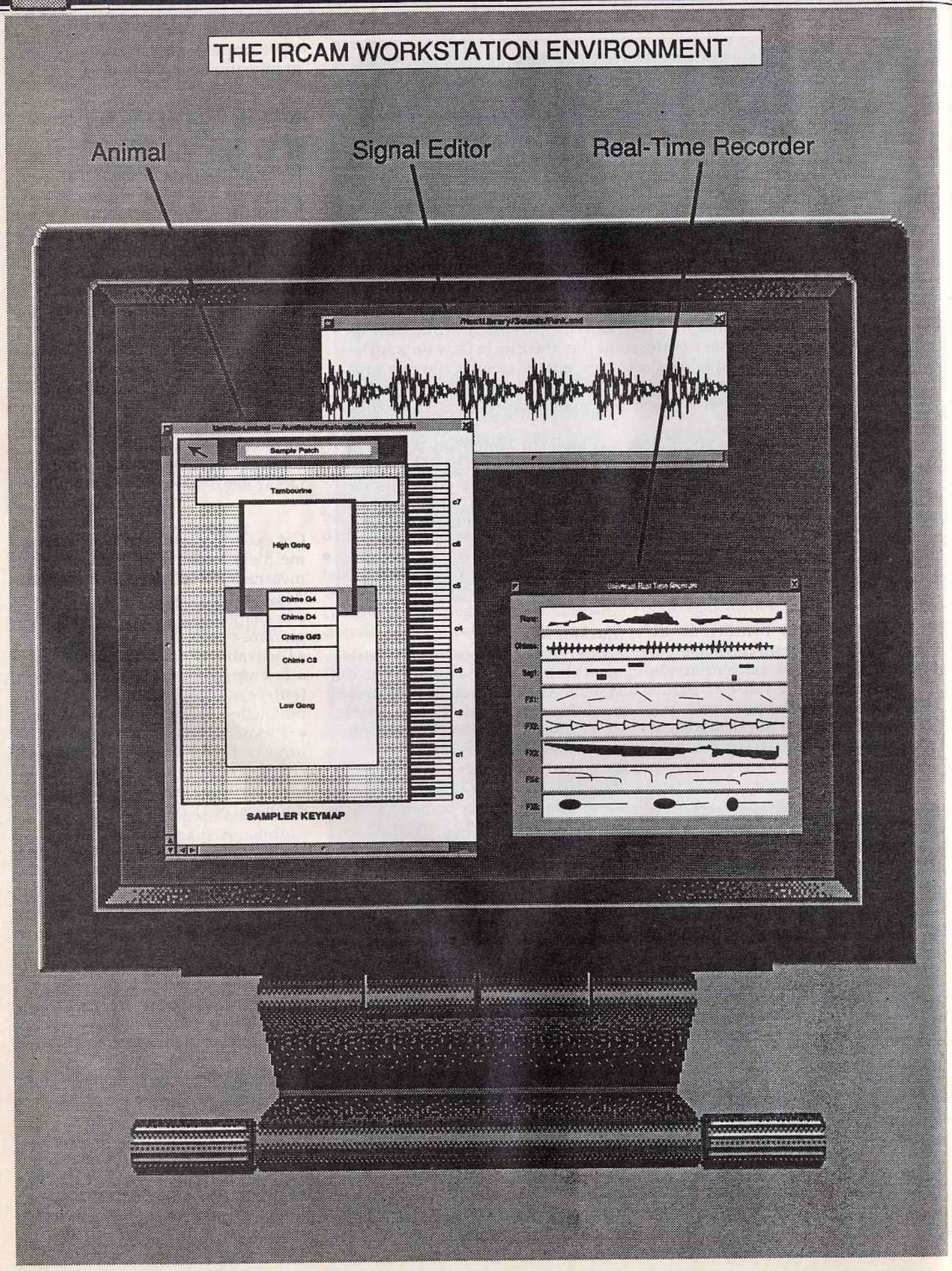
- le chargement de modules de programme ;
- l'activation d'objets (instanciation) ;
- la communication d'arguments (entrée/sortie) lors de l'exécution des méthodes appliquées aux objets ;
- l'exécution, à la requête du noyau IMRTX, de procédures UNIX (création, ouverture/fermeture, écriture/lecture de fichiers de données, etc.).

Un mécanisme de chargement dynamique permet, par ailleurs, d'intégrer, au sein de l'image du système en cours d'exécution, de nouvelles définitions d'objets.

IMRTX met en oeuvre deux types de mécanismes fondamentaux, essentiels pour l'accomplissement des tâches qui lui sont assignées : la prise en compte, en temps réel, des événements survenant dans le cadre de l'exécution, et le traitement de flots de données à caractère synchrone.

Gestion généralisée d'événements

Le mécanisme d'interruption est utilisé pour prendre en compte les événements déclenchés, de manière aléatoire, par l'environnement interactif (sollicitation de l'utilisateur, intervention d'un temporisateur,


THE IRCAM WORKSTATION ENVIRONMENT
Animal
Signal Editor
Real-Time Recorder
Sample Patch
Tambourine
High Gong
Chime G4
Chime D4
Chime G#3
Chime C3
Low Gong
c7
c6
c5
c4
c3
c2
c1
c0
SAMPLER KEYMAP

etc.). Le traitement de ce type d'interruption peut impliquer des actions complexes :
- acquisition de ressources (gestion de la mémoire, accès disque, etc.) ;
- activation/désactivation d'objets (initialisation/libération) ;
- tris, recherche dans des listes ou tables de données ; etc.

Afin d'uniformiser la programmation du système, le code des gestionnaires d'interruption adopte une formalisation de type objet : les événements sont des messages expédiés à des objets, lesquels sont à leur tour susceptibles d'expédier des messages à d'autres objets (eux-mêmes y compris).

Traitements synchrones

Les opérations typiques effectuées en traitement de signal portent en majorité sur l'établissement et la transformation ordonnée de flots de

"Le principe de cet exécutif est de faire tourner, en mode monotâche, un processeur exécutant une boucle de traitement d'événements. C'est la structure même d'un programme Macintosh : l'application est, non pas une boucle, mais un programme passif qui capte les événements, et les range dans une liste. Il n'y a pas d'interruption, ou mille tâches différentes, mais une seule tâche qui traite un événement, puis va chercher l'événement suivant, et ainsi de suite.

"En ce qui nous concerne, chaque échantillon, ou plus exactement chaque vecteur d'échantillon est traité par interruption. En audionumérique de qualité professionnelle, l'échantillonnage s'effectue toutes les 22,6 microsecondes, si bien qu'en travaillant avec des vecteurs de 32 échantillons, l'interruption intervient toutes les 725 microsecondes. A ce moment-là, il faut exécuter une suite de fonctions liées au traitement de signal (il y a une liste de fonctions, non-conditionnelles, tournant en permanence), puis après avoir effectué ce travail, le processeur consulte ses listes d'événements, pour savoir si des événements de type aléatoire ont été déclenchés dans cette fenêtre de temps (frappe du clavier, clic de la souris, etc.). Si c'est le cas, il procède au traitement adéquat.

"Dans ce contexte de traitement de signal, une optimisation importante peut être accomplie, à cause du déterminisme du problème. Ceci n'est pas possible avec les autres modèles d'exécutif temps réel."

Eric Lindemann

12 - 13 - 14 Septembre
PARIS

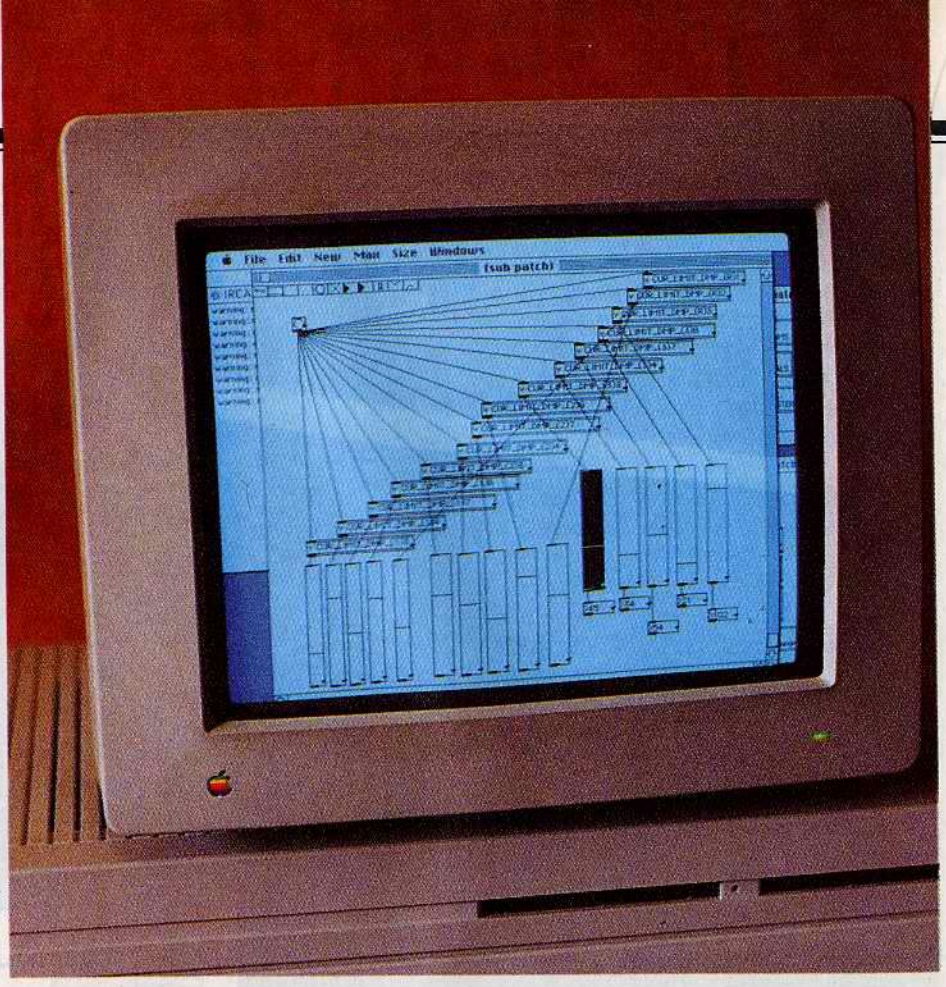
..... UNIX .....
.... MICRO ....
..... MINI .....
SYSTEMES OUVERTS
.. PORTABILITE ..
.... GESTION ....
. APPLICATIONS .
.. BUREAUTIQUE ..
..... PICK .....
.... RESEAUX ....
.....

SOLUTIONS !

Le Salon des Systèmes ouverts et des Applications Informatiques sous UNIX

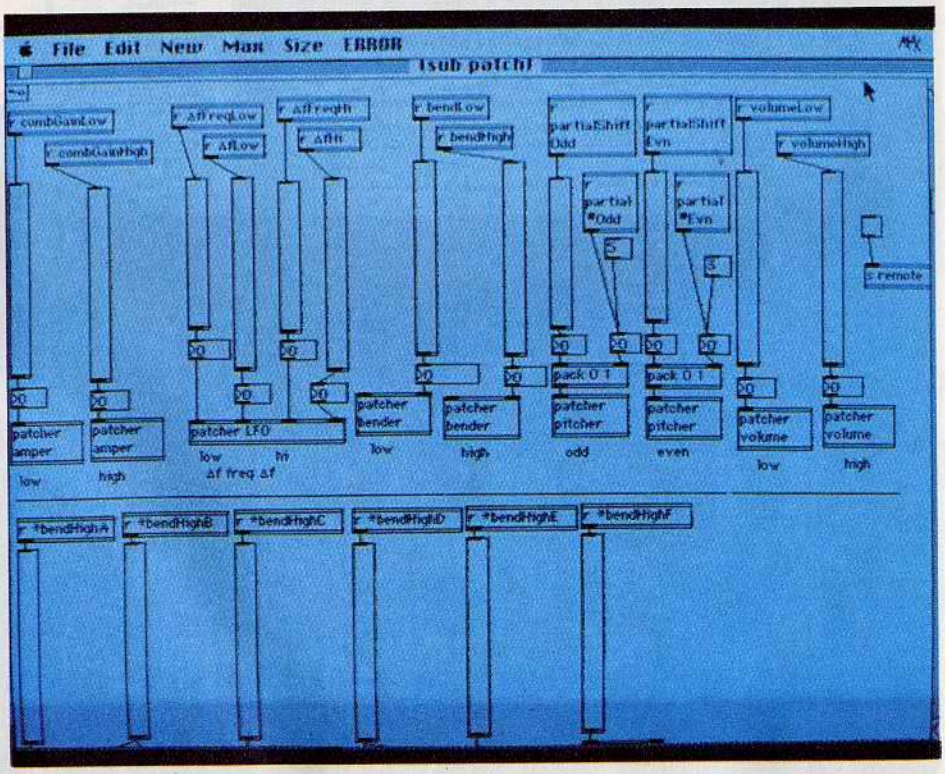
INFOPROMOTIONS : 15/17, avenue Ledru-Rollin - 75012 PARIS - Tél. : (1) 43.44.35.97

données : les composantes du signal à analyser/modifier, sur une base temporelle régulière (dont la fréquence est déterministe). Ainsi le déclenchement d'une opération peut comporter des calculs portant sur les millions de données composant un échantillon sonore. Dans la plupart des cas, les calculs sont répartis sur plusieurs microprocesseurs i860 travaillant en parallèle. Ils constituent les noeuds d'un réseau de traitement de flots de données synchrones (SDN : *Synchronous Dataflow Network*). Les algorithmes des routines qui implémentent ces traitements adoptent, le plus souvent, des structures relativement simples et aisément parallélisables (peu d'opérations conditionnelles, opérations vectorielles, etc.), dont le temps d'exécution est prévisible. Ces caractères facilitent la mise en correspondance, de manière optimale, de la structure des SDN avec l'architecture multiprocesseur du système.



Cohérence temporelle

Les événements externes sont estampés par un processus de datation qui détermine, par rapport à un moment de référence, le moment précis de leur occurrence. Cet estampage est transmis tout au long de la chaîne de communication entre objets, activée lors du traitement d'un tel événement. Or, la structure (pipeline) de la transmission des messages entre les noeuds du réseau, lors de la parallélisation des calculs déclenchés par cet événement, entraîne des délais qui doivent être pris en compte. Le graphe d'un SDN est donc pondéré par ces délais afin que, quelle que soit sa topologie, la modification d'un noeud (modification du coefficient d'un filtre, par exemple) ne détruise pas la cohérence de l'ensemble du traitement à opérer.



Une constante du système est associée à la durée du traitement d'un événement. Correspondant au temps de diffusion écoulé entre l'occurrence de celui-ci et le moment où la frontière du SDN est atteinte, elle est configurable par l'utilisateur dans une plage allant de 1 à 10 ms. Les messages parvenant au SDN endeçà de la valeur fixée sont retardés, alors que ceux qui parviennent audelà sont pris en compte dès que les traitements en cours le permettent. Grâce au positionnement d'un indicateur, l'utilisateur est averti de cette acceptation.

SDN concurrents

L'activation de chaque SDN étant autonome, on peut obtenir un recouvrement partiel ou total de plusieurs SDN, dont les moments d'activation et de terminaison s'échelonnent dans le temps. Chaque SDN est affecté à une "fenêtre" de potentiel de calcul représentant une fraction de la puissance de calcul globale du système. Il est même possible de faire coexister plusieurs SDN hétérogènes dans une même fenêtre. Lors de l'allocation d'un SDN à une fenêtre, la mémoire

LE MICROPROCESSEUR INTEL i860

Présenté par Intel, lors de son lancement au début de l'année 89, comme microprocesseur de type RISC destiné à être employé comme unité centrale de traitement dans un système informatique, la mise en situation du i860 a montré qu'il devait être plutôt rattaché à la catégorie des coprocesseurs de calculs mathématiques et traitements graphiques. C'est, de fait, la position qu'il occupe dans le système développé à l'IRCAM, où ses capacités sont utilisées de façon optimale.

L'architecture fonctionnelle du i860 repose sur trois unités de calcul indépendantes, opérant en parallèle :
- unité de calculs entiers (IU : *Integer Unit*) ;
- unité de calculs flottants (FPU : *Floating Point Vector Unit*) ;
- unité de traitements graphiques (GU : *Graphics Unit*).

Trois autres unités fonctionnelles, dédiées aux transferts des données complètent ce groupe de calcul :
- unité de gestion de mémoire virtuelle et d'interfaçage du bus (MMU : *Memory Management Unit*), assurant une bande passante sur le bus de 160 Mo/s ;
- caches d'interfaçage entre le microprocesseur et la mémoire centrale (suivant le schéma de Harvard), assurant une bande passante totale de 900 Mo/s (320 + 640 Mo/s, version 40 MHz) :
  - mémoire d'instructions (4 Ko) - chemin d'accès de 64 bits ;
  - mémoire de données (8 Ko) - chemin d'accès de 128 bits.

Le i860 est pourvu d'un fichier de registres comportant :
- 32 registres (32 bits) utilisés par l'IU ;
- 16 registres (64 bits) utilisés par la FPU et la GU ;
- 4 registres spéciaux (64 bits) utilisés par la FPU et la GU ;
- 6 registres de contrôle/état (32 bits) ;
- 1 registre pointeur d'instruction (32 bits).

Le i860 est échantillonné en deux versions, correspondant à des fréquences d'horloge de 33 et 40 MHz.

Unité entière

L'IU comporte un noyau de type RISC (jeu réduit de 38 instructions), véritable centre nerveux du i860, prenant en charge :
- l'extraction des instructions destinées à :
  - l'IU elle-même ;
  - la FPU;
- le décodage et l'exécution des instructions de l'IU :
  - arithmétiques ;
  - logiques ;
  - branchements ;
  - trappes ;
- le chargement/rangement des opérandes (IU et FPU) ;
- le calcul des adresses fournies simultanément aux caches (instructions/données) ;
- l'exécution des instructions de rangement de pixels, dans les calculs d'élimination de surface cachée effectués par la GU ;
- le positionnement de l'état du processeur (registres de contrôle).

L'utilisation des registres est optimisée par la présence d'un registre de marquage (*scoreboard*) des registres occupés (destinataires d'un calcul en cours).

L'IU travaille à un rythme d'exécution d'une instruction par cycle.
Performance :
80 KDhrystones v2.1 (33 MIPS VAX).

Unité flottante

La FPU accepte un jeu de 18 instructions, opérant sur des données conformes au standard IEEE-754 (simple et double précisions - SP/DP). Elle se décompose en deux modules, capables de travailler en parallèle, entre elles et avec l'IU :
- additionneur ;
- multiplicateur.

Ces deux modules découpent l'exécution des instructions en plusieurs étapes (pipeline à 3 étages). Toutefois, il est possible de forcer l'exécution de certaines opérations en mode "continu".

Le rythme d'exécution s'établit à :
- 3 cycles (additions SP/DP, multiplication SP) ;
- 4 cycles (multiplication DP).

Il faut préciser que la FPU ne comporte pas d'instruction de division.
Performances :
24 MWhetstones
10 MFLOPS Linpack
FFT(1024 points) en 1 ms.

Unité graphique

La GU est articulée autour d'un jeu comportant 12 instructions permettant d'implémenter les primitives graphiques du type :
- élimination de surface cachée ;
- interpolation de distance ;
- ombrage 3D (interpolation de l'intensité des pixels) - 48.000 polygones ombrés/s (avec contrôle de *Z-buffer*).

Ce jeu inclut, entre autre, des instructions arithmétiques (addition et soustraction) portant sur des opérandes entiers de 64 bits.

Comme dans la FPU, les instructions peuvent passer (ou non) par les étages d'un pipeline. La GU fait d'ailleurs usage des registres généraux de la FPU pour ses calculs, ainsi que d'un de ses registres spéciaux (*special merge register*). Elle peut travailler en parallèle avec l'IU.

Unité de gestion mémoire

Adoptant le principe de mémoire paginée (taille de page fixé à 4 Ko), elle fonctionne par demande de page. La répartition des pages s'effectue au travers de deux niveaux de translation, dans un espace virtuel de 4 Go. La base de la table des pages est pointée par un registre de contrôle (*Directory Base Register*). L'opération de translation d'adresse virtuelle en adresse physique fait usage d'un tampon de translation à 64 entrées et s'effectue en un cycle, en parallèle avec l'accès des caches d'instructions/données.

Le microprocesseur i860 a fait l'objet d'une définition d'interface d'application binaire (ABI : *Application Binary Interface*). Celle-ci garantit la compatibilité des systèmes qui l'utilisent dans le cadre d'UNIX System V révision 4.

"Si les noyaux temps réel conventionnels sont optimisés pour la gestion d'événements dans des systèmes acceptant de nombreuses interruptions, ils ne le sont pas suffisamment en ce qui concerne le traitement de signal. Sur les processeurs habituels, le temps de prise en compte des interruptions, pour un bon noyau temps réel, est d'environ 20 à 30 microsecondes.

"Or, nous avons besoin de descendre en-dessous d'une microseconde, ce qui implique de traiter l'interruption de l'horloge du traitement de signal très efficacement : presque sans (ou avec très peu de) délai au niveau système - dans ce cas particulier d'interruption effectuant le traitement d'échantillons audionumériques (tous les autres événements étant récupérés sur cette interruption dans la liste des messages).

"Nous n'avons pas trouvé sur le marché ce que dont nous avions besoin : un exécutif temps réel mélangeant traitement de signal et gestion d'événements, compatible, de surcroit, avec le monde Unix, ceci afin de pouvoir, dans un contexte de recherche, lancer un programme exécutant des appels système (*fork*, *sbreak*, *fclose*, etc.). Nous avons donc décidé de tracer notre propre chemin : construire un système d'exploitation temps réel tournant sur les processeurs Intel i860, système multitâche, multiprocesseur, optimisé pour le temps réel."

Eric Lindemann

nécessaire à son fonctionnement est réservée et initialisée - mais le SDN n'est toutefois pas encore activé. Si la puissance de calcul correspondant à une fenêtre le permet, plusieurs de ces SDN peuvent être activés concurremment. Ce qui n'empêche nullement que plusieurs fenêtres soient actives simultanément. En cas de conflit provoqué par le dépassement de la capacité de calcul du système, IMRTX possède les mécanismes qui lui permettent de "préempter" un ou plusieurs SDN afin d'accorder la priorité à un SDN plus important.

ENVIRONNEMENT LOGICIEL

Pour pouvoir répondre, non seulement aux besoins internes de l'IRCAM (développements prévus dans le cadre des activités pédagogiques, applications entreprisent pour la réalisation de commandes institutionnelles, etc.), mais aussi à ceux de sociétés logicielles souhaitant créer des applicatifs destinés au marché scientifique ou artistique, le projet de station musicale incorpore plusieurs couches logicielles d'assistance au développement.

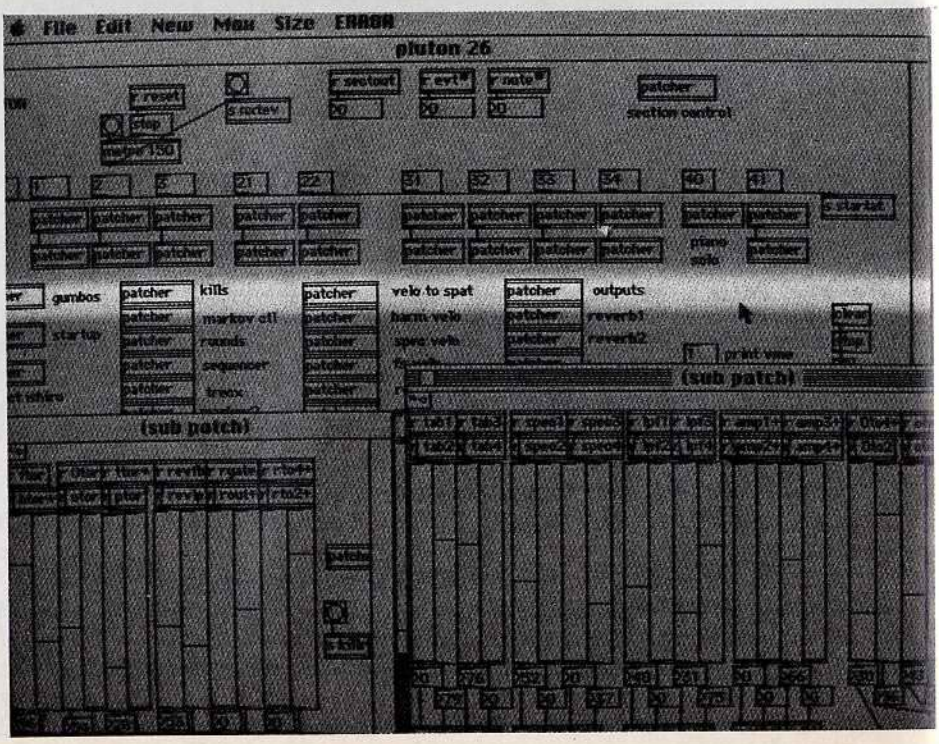
Animal

Pour les programmes élaborés dans le contexte de l'exécution temps réel, l'interface homme/machine représente une des parties les plus importantes des efforts de programmation. Animal est une couche logicielle ayant pour but de faciliter la mise au point des interfaces de ces programmes, en proposant :

- des objets graphiques de base, adaptés à la représentation et au contrôle de données évoluant dans un contexte dynamique temps réel ;
- une organisation et une représentation hiérarchique des objets contenant les informations à traiter ;
- des espaces de travail définissables par l'utilisateur, permettant la manipulation de collections d'entités graphiques ;
- la définition de relations entre objets, sous forme de connexions graphiques ;
- un interfaçage simplifié avec un environnement de programmation évolué (langage de haut niveau).

Animal constitue en fait un arbre de fenêtres. Chacune d'elles peut contenir un ou plusieurs "graphiques" comportant un nombre quelconque de parties mobiles. Chacune donne accès à sa propre fenêtre subordonnée. Un "animal" correspond à cet assemblage d'un graphique et de sa fenêtre subordonnée, soit, en terme d'interface, aux positions et dimensions des parties mobiles du graphique, et aux pointeurs sur les animaux définis dans la fenêtre subordonnée. A chaque animal sont attachées des "méthodes" (fonction, dans la terminologie de la programmation par objets) qui décrivent les actions (les messages) auxquelles il répond. Le programme Animal étant conçu pour fonctionner sur une architecture multiprocesseur, les méthodes affectant un animal peuvent s'exécuter sur des processeurs différents de celui qui exécute la tâche du graphique. Le langage C++ est utilisé pour réaliser leur implémentation : dans la formalisation de ce langage, la définition d'un animal correspond tout simplement à une classe.

Grâce à la formalisation adoptée, il est aisé de réaliser en plus de la sauvegarde sur disque de la hiérarchie graphique complète d'une application, la sauvegarde individuelle d'un graphique, afin de constituer des bibliothèques d'animaux (*sic*) réutilisables.



MAX

Ce logiciel appartient à une génération d'outils de composition musicale issue des travaux menés ces dernières années à l'IRCAM. Apparenté aux outils de programmation graphiques utilisés en génie logiciel, il utilise le formalisme d'objet pour décrire les matériaux sonores et leurs transformations.

Avec le logiciel Max, l'exécution musicale consiste à mettre en oeuvre un ensemble d'objets établis par le compositeur, dont les caractéristiques et le comportement sont décrits sous forme de classes (on peut assimiler celles-ci à des "moules" servant à créer les objets nécessaires à l'élaboration du matériau musical). Afin de doter ce modèle de capacités évolutives, le système organise les interactions entre les objets : l'échange des informations repose sur des mécanismes de transmission de messages, indiquant les actions à exécuter et les arguments qui s'y rapportent.

Les principales classes réunies dans Max sont du type :
- fenêtre ;
- éditeur de texte ;
- éditeur de tables de fonctions ;
- composant élémentaire (*patcher*) ;
- séquenceur.

Des fonctions systèmes de base prennent en charge la temporisation des événements, les E/S MIDI, la gestion de la mémoire, et la transmission des messages.

Une extension du programme Max (initialement appelée *Patcher*) permet de réaliser en temps réel la configuration et l'activation des objets produits dans l'environnement Max, au travers d'une représentation entièrement graphique. Chaque objet dispose de points d'entrée (*inlets*) et de sortie (*outlets*) qui permettent de figurer, par des liaisons linéaires, les relations qui l'unissent aux autres objets. L'ordonnance de l'activation des objets prend ainsi l'apparence d'un diagramme de flot de données, comparable à ceux que produisent, en génie logiciel, les méthodologies d'analyse structurée. Le système est hiérarchique et permet l'abstraction, sous la forme d'un objet de niveau supérieur (pourvu de ses propres *inlets* et *outlets*), d'un groupe d'objets ordonné.

Trois types fonctionnels catégorisent les objets de Max, représentés graphiquement sous forme de boîtes :
- contrôles (boutons de déclenchement, potentiomètres linéaires, indicateurs de valeur numérique, etc.);
- E/S (les *inlets* et *outlets* d'un objet de niveau hiérarchique supérieur);
- textes (messages, commentaires, création d'objet).

Le comportement des objets est réglé par un ensemble de messages, l'équivalent, en programmation par objets, des "sélecteurs" correspondant aux "méthodes" définies pour chaque classe d'objets. Ces messages sont diffusés le long des liens établis entre les différents objets. On trouve des messages standards tels que :
- *bang* (déclencheur) ;
- *fix* (indicateur de nombre en virgule fixe) ;
- *float* (indicateur de nombre en virgule flottante).

D'autres types d'objets servent pour :
- les opérations mathématiques (opérations de base, comparaisons, opérations trigonométriques) ;
- les fonctions de filtrage (sélections sur valeur non nulle ou déterminée, etc.) ;
- les traitements des informations MIDI (séquenceur, suiveur de partition, conversions de flots de données MIDI) ;
- les fonctions de temporisation (retard, métronome, décomptage, analyse de tempo, etc.) ;
- des fonctions diverses (contrôles de messages, de séquencement, etc.).

Max constitue un outil interactif parfaitement adapté à la gestion de l'exécution en temps réel, dont la puissance et la souplesse permettent l'utilisation dans des environnements et dans le cadre de styles créatifs très variés. Il facilite la représentation des événements musicaux et leurs apparitions sous l'aspect de "partition virtuelle", dans le contexte d'un dialogue entre instrumentiste/ordinateur. Celui-ci peut, par exemple, prendre la forme de l'exécution d'une partie autonome, calculée, en temps réel, en appliquant des fonctions statistiques (chaînes markoviennes) à l'analyse du jeu d'un instrumentiste.

Le compositeur Philippe Manoury a récemment mis ce logiciel en situation, dans une pièce (*La Partition du ciel et de l'enfer*) réunissant une flûte MIDI (1), un piano MIDI, le système 4X, et une formation instrumentale. Elle met en oeuvre une méthode de synthèse nouvelle, créée et implémantée par Miller Puckette dans l'environnement Max, qui permet de contrôler la fréquence, les composantes spectrales du timbre, les composantes bruités, et le vibrato d'un son.

Le développement de Max, qui se poursuit actuellement, ainsi que son portage sur la station musicale IRCAM (NeXT), doit déboucher sur une prochaine commercialisation par la firme Opcode, exemple de la nouvelle politique de diffusion entreprise par l'IRCAM.

(1) Cet instrument, mis au point à l'IRCAM, utilise des capteurs pour la transformation des signaux en informations codées au format MIDI.

Références :

Miller Puckette : The Patcher, in Proceedings ICMC, Köln '88.

Prochainement dans le cahier Technologies Avancées :
Architectures et langages du parallèlisme,
Réseaux neuronaux...

Enregistreur universel
Recouvrant les fonctionnalités d'un enregistreur de données multipiste, ce programme effectue l'enregistrement, en temps réel, des données pouvant correspondre à n'importe quel type de formatage (signaux audio, événements MIDI, structures définies par l'utilisateur, etc.). Plusieurs pistes enregistrées sont éditables concurremment, chacune d'elles employant l'outil adéquat (éditeur de signal, d'événements, etc.).

Editeur de signal
En mettant en oeuvre les éléments fournis par le programme Animal, un éditeur spécifique adapté à la représentation interactive des signaux audio, tant du domaine des fréquences que du temps a été intégré à l'environnement de développement.

Bibliothèque de traitement de signal
Une bibliothèque de routines de traitement de signal optimisées, utiles dans le contexte de la station musicale, vient compléter les outils développés autour de la station de l'IRCAM. En plus des fonctions mathématiques standards (addition et multiplication vectorielles, multiplication matricielle, FFT, filtre de second ordre, etc.), elle contient des algorithmes nécessaires à la programmation des primitives d'un système de synthèse sonore informatisé (oscillateur à table d'interpolation, etc.).

ORIENTATIONS TECHNOLOGIQUES

En intégrant délibérément des éléments situés à la pointe de la technologie, autant sur le plan matériel que logiciel, la station musicale en cours de réalisation doit marquer une étape importante dans l'évolution de la micro-informatique musicale. La détermination de ses spécifications a été effectuée en tenant compte des tendances actuelles du marché informatique. Ouverture et standardisation s'inscrivent, en effet, parmi les principaux critères de sélection des composantes du projet initial.

Ouverture, en choisissant la plate-forme NeXT. Celle-ci est effectivement reconnue la plate-forme de choix pour des développements dans le secteur de l'audionumérique (rappelons que le NeXT est actuellement le seul micro-ordinateur intégrant un processeur de traitement de signal, celui que l'on retrouve d'ailleurs sur la carte réalisée par l'IRCAM : le DSP56001). L'apparition de périphériques de ce type ne peut que renforcer l'intérêt du système proposé.

Standardisation également, en adoptant le monde UNIX comme environnement du système, le meilleur garant de son universalité d'emploi et de son évolutivité.

Ajoutons, avant de conclure, que la station musicale de l'IRCAM est appelée à suplanter le système 4X, en tant que système temps réel principal, ceci à l'horizon 91. A cette date, le nouveau système devrait entrer en phase de commercialisation, par l'entremise de la firme américaine Ariel, qui s'illustre tout particulièrement par le développement d'outils et de périphériques de traitement du signal (microphone numérique pour le NeXT, cartes et outils de développement pour le DSP56001, etc.).

Daniel Fournier

VOUS AIMEZ LES

TECHNOLOGIES AVANCEES?

VOUS ETES LE BIENVENU SUR LE

3615 STMAG

TAPEZ *UNIX OU *TEC PUIS ENVOI

CHANT, FORMES : SYNTHESE

Synthétiser la voix chantée a été l'un des premiers projets mis en chantier à l'IRCAM. Très vite, il a donné naissance à une famille d'outils de base qui sert désormais de référence.
Xavier Rodet, Responsable du Département Analyse-Synthèse et initiateur de ces travaux, présente les aspects essentiels des techniques mises en oeuvre.

CHANT

Dans la synthèse de la parole à partir de texte - le thème de recherche auquel je consacrais ma thèse d'État dans les années 75 -, le principe de construction du signal consistait à assembler de petites formes d'ondes représentant des formants - éléments essentiels de la reconnaissance et de la synthèse de la parole. En choisissant cette technique de séparation, il devenait possible de manipuler individuellement, et donc de faire varier les fréquences des différents formants.

La qualité du résultat fut très satisfaisante, et, fait inattendu, le temps de calcul étonnament réduit par rapport aux techniques concurrentes, permît d'effectuer une synthèse quasiment en temps réel. Les compositeurs travaillant à l'IRCAM furent particulièrement intéressés par ces résultats, notamment parce que, pour eux, les formants sont conceptuellement très "parlants" : fréquence et timbre repérables, décroissance naturelle du son, etc. Ce sont ces idées de base que le programme *Chant* a reprises : utiliser une description en terme de mode de résonance, fréquence, taux de décroissance - en fait la largeur de bande du filtre équivalent - et caractériser le mode et la vitesse d'excitation.

Avec *Chant*, les compositeurs apprécièrent de pouvoir trouver une correspondance directe entre les paramètres et les faits sonores, au lieu d'avoir à manipuler une sinusoïde modulée par un index. De plus, ayant en fait réalisé la modélisation d'un phénomène physique, le principe pouvait s'appliquer à la plupart des instruments, dont le fonctionnement est analogue, suivant le modèle excitation/résonance. Une seule technique de synthèse permettait ainsi de synthétiser des trompettes, des violons, des cloches, et même des choses tout à fait incongrues : des cymbales ou des grosses caisses ! Avec, en prime, une qualité de son bien plus riche, bien plus subtile que celle que l'on obtenait avec les autres méthodes de synthèse.

PRINCIPES

Cette technique des formes d'onde formantiques consiste à calculer les signaux des modes de résonance, chaque mode de résonance étant considéré comme un signal indépendant, à savoir une sinusoïde affectée d'une enveloppe décroissante exponentielle. Pourtant, l'attaque du son n'est pas très exactement le point de départ de l'exponentielle, mais une sorte de lissage initial dont le rôle est très important : il représente la vigueur de l'attaque - une attaque extrêmement brutale est représentée par une discontinuité initiale à partir de l'exponentielle, et, au contraire, une attaque très douce par un lissé très progressif précédant la décroissance exponentielle.

L'implémentation algorithmique et la programmation de ce modèle ont été facilement effectuées : il a eu, entre autres, des implantations sur des transputers Inmos, sur des TMS 320, etc. A l'IRCAM, la première

version fut réalisée sur un PDP-10 à l'aide de SAIL (*Stanford Artificial Intelligence Language*), une extension d'Algol (1).

FORMES

Mais pour que le son soit utilisable musicalement, il est nécessaire d'effectuer une bonne description des paramètres qui contrôlent le synthétiseur. Toute une partie de notre travail a donc consisté en une acquisition des paramètres - d'après l'analyse des signaux - et en une modélisation de leur variation au cours du temps. Or, ces paramètres sont très complexes (dynamisme du jeu, du chant, ou de la percussion ; type d'articulation : legato, portamento, staccato, etc.). En conséquence, les programmes d'analyse sont extrêmement délicats à mettre en oeuvre, car il faut donner aux utilisateurs des leviers de commandes maîtrisables.

Lorsqu'il a fallu appliquer les règles établies et transcrire les algorithmes en langage, la notion d'objet s'est très vite imposée, ou plus exactement de processus, car la notion de temps intervenait de façon essentielle. Un processus est, en fait, une entité informatique qui exécute un certain algorithme, dans un contexte donné, en un temps donné, et pour une durée donnée. D'autre part, la structure du phénomène musical peut être, en grande partie, décrite par des hiérarchies, des juxtapositions et des parallélismes : phrase découpée en motifs, regroupant des notes, décomposables en attaque - entretien - décroissance, etc.

Progressivement, nous avons construit un système assez complexe appelé *Formes*, regroupant à la fois une grande bibliothèque de règles (vibrato, portamento, crescendo, fonctions élémentaires : passer de décibel à amplitude linéaire, de note représentée à une échelle logarithmique à la fréquence fondamentale correspondante, etc.), la notion de processus déclenchant ces règles, dans des limites temporelles, et toute la structure hiérarchique des processus (processus décrivant l'attaque, la note, un autre groupant plusieurs notes en un motif, etc.).

IMPLEMENTATION

La mise en oeuvre interactive de ce type de processus fut réalisée en Lisp (2) et depuis des années, le système *Formes* a été utilisé par de nombreux compositeurs. Depuis 1986, la nouvelle génération représentée par *PréFORM* est venue ajouter une boîte à outils donnant accès au temps réel, à un langage à objets complet, au graphisme, à l'interactivité, etc., et suscite des développements fondamentaux importants (*Esquisse*, *Patcher*, etc.).

Dans la partie analyse, nous représentons un signal - voix ou instrument - par ses caractéristiques acoustiques, à l'aide de la bibliothèque UDI (*Unified DSP Interface*) - un protocole de traitement de signal que nous avons développé. La partie synthèse regroupe des programmes spécifiques, réalisant différentes techniques de synthèse à partir des paramètres d'analyse. La plus importante d'entre elles fonctionne sur le principe source/filtre - comme dans *Chant* -, car les modes de résonance peuvent être vus comme la résonance d'un filtre.

Pour déterminer les paramètres pertinents du signal, nous utilisons deux approches complémentaires : modèle de production et modèle de perception. Les paramètres importants sont ceux que nous entendons, et il est donc inutile de recomposer les hyperfréquences que nous ne percevons pas. Parmi les paramètres qui ont un effet direct sur notre écoute, l'enveloppe du spectre est responsable de la perception phonétique du son. Si on remplit l'enveloppe spectrale d'une voix avec n'importe quelle structure, la structure phonétique est quand même perçue. Il en est de même pour les instruments : la forme de cette enveloppe détermine des aspects - brillance plus ou moins accentuée, équilibre du son, etc. - qui constituent une grande partie du timbre.

L'autre approche (la structure fine du spectre) détermine la fréquence fondamentale - par l'espacement des raies -, la présence de bruits par rapport à des composantes de fréquence bien définie, et les mouvements dynamiques de ces composants.

OUTILS

Les outils d'analyse que nous avons construits donnent donc l'enveloppe spectrale et les caractéristiques fines du spectre. Ce sont des modèles constamment améliorés pour être de plus en plus précis et robustes, tout un ensemble très volumineux de logiciels délicats à mettre au point.

A chaque technique d'analyse correspond, en général, une technique de synthèse, représentée par un programme tout à fait nouveau, mais qui, d'une certaine façon, inverse le programme d'analyse. Il est même possible d'envisager des croisements entre plusieurs techniques d'analyse et de synthèse, démarche extrêmement féconde, à condition que les interfaces entre ces différentes techniques soient possibles. Le problème est de normaliser des choses qui, a priori, ne sont pas équivalentes. Par exemple, la prédiction linéaire - une technique d'analyse d'enveloppe spectrale utilisée notamment avec la parole - est un moyen d'obtenir l'enveloppe spectrale. Les paramètres de celle-ci sont, en fait, les paramètres d'un filtre, mais ils sont inutilisables dans le cas d'une synthèse additive ou FM. Il faut alors les convertir, non seulement en terme de traitement de signal, mais aussi de format de fichier.

Un de nos objectifs sera de disposer, sur les futures stations musicales de l'IRCAM, de tous ces outils, avec la possibilité de passer d'un programme d'analyse à un programme de synthèse presque indifféremment.

Propos transcrits par
Daniel Fournier

(1) Langage comparable au C, et même plutôt meilleur !

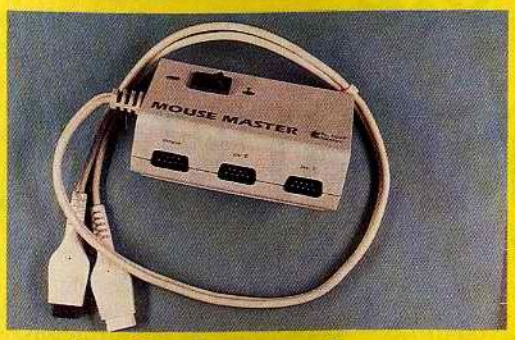
(2) Un des langages les plus efficaces, en terme de facilité d'utilisation et de vitesse d'exécution.

OFFRE N°1

MOUSE MASTER

Boîtier de raccordement avec un double port souris commutable à l'aide d'un bouton et un port joystick (3 prises au total)

290 francs + 15 Frs de port



Le CD des musiques du jeu "ROCK STAR"

65 francs l'unité (port compris). GENIAL !

OFFRE N°2

AVEZ-VOUS LA COLLECTION COMPLÈTE DE ST MAG ??

☆ LE PACK DU SIÈCLE N° 1 ☆

(Barrez les numéros inutiles)

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

ATTENTION !Certains numéros indiqués ne sont pas encore sortis. Ne commandez que les numéros précédant celui du mois en cours.

Il comprend :

4 RELIURES ou COFFRETS (entourez votre choix) + 28 ANCIENS NUMÉROS AU CHOIX :

650 Frs (port compris)

Une occasion à ne pas manquer, la valeur réelle du pack est de 1060 francs !!!

OFFRE N°3

RELIURES ET COFFRETS

Tous deux sont bleu roi et frappés du sigle ST MAG. Les coffrets permettent de ranger votre collection bien au chaud, à l'abri des tremblements de terre et des raz de marée. Les reliures permettent comme leur nom l'indique de relier votre collection comme des pros. Chacun de ces indispensables investissements peut contenir 10 numéros.

Le coffret ou la reliure ST MAG :

65 Frs pièce port compris.

BOITE À FICHES 22.60 francs

couleur argent, or ou bleue métallisée

OFFRE N°4

3 MANETTES EN UNE SEULE

Une souris (solide) + un Track ball (rapide) + un joystick à 345 Ff +15ff de port

Recommandé par le magazine, pratiquement incassable, c'est l'accessoire indispensable.

Super Nouveau Chouette Facile C'est possible...

Désignation	Qté	P.U.	TOTAL
1 MOUSE MASTER		290	
Le CD du jeu "ROCK STAR"		65	
2 Le Pack du siècle		650	
3 La reliure ST Magazine		65	
La boite à fiches		22,60	
4 Le Track Ball 3.0		345	
TOTAL GÉNÉRAL			

NOM : .....

Prénom : .....

Adresse : .....

Code Postal : .....

Ville : .....

Merci d'envoyer votre règlement à l'ordre de PRESSIMAGE, 210 Rue du Fbg St Martin 75010 PARIS, par CCP ou chèque bancaire. Etranger : virement bancaire ou chèque encaissable S.P.

PEDAGOGIE DE L'INFORMATIQUE MUSICALE

A partir de la prochaine rentrée universitaire, l'IRCAM intensifie ses activités pédagogiques en instaurant un DEA d'informatique musicale. Jean-Baptiste Barrière, Responsable du Département Pédagogie de l'IRCAM, en trace les lignes directrices, et évoque les problèmes posés par la formation des compositeurs aux nouveaux moyens d'expression.

Quel va être le contenu du Cursus d'Informatique Musicale ?

Il va couvrir une année de formation extrêmement intensive et très lourde : les participants devront y consacrer tout leur temps et même plus ! Il s'adresse à des musiciens ayant terminé leur formation - des compositeurs que l'on peut déjà distinguer d'après leur travail - mais qui, sur le plan de l'informatique musicale, n'ont pas encore reçu de formation - ou dont la formation a été plutôt empirique.

Au cours de la première partie du Cursus, une partie théorique qui s'étendra sur neuf mois, les participants recevront une initiation complète à la programmation en C et en Lisp. En observant dans le détail ces deux langages assez radicalement différents, l'objectif sera d'abord de les rendre capables de programmer des choses assez élaborées, mais surtout de lire des programmes. Cette initiation s'appuiera uniquement sur des exemples musicaux, et même dans les cas les plus éloignés, l'application musicale restera sous-jacente. En plus de cette partie programmation, ils examineront tous les concepts qui agitent l'informatique - jusqu'à l'intelligence artificielle et les réseaux neuronaux, par exemple.

Il y aura également un séminaire scientifique, qui se fixera pour objectif d'étude - en adoptant une méthode analogue à celle qui est appliquée dans l'étude de la programmation - les concepts, les notions, les thèmes de connaissance qui articulent la formation musicale. Car la programmation est un aspect particulier sur lequel il faut insister, mais il y a beaucoup d'autres sujets à considérer : l'acoustique, le traitement de signal, etc.

Ensuite, il y aura une partie informatique musicale - au sens strict du terme - qui recouvrira tous les thèmes que cette discipline aborde aujourd'hui : de l'analyse des sons par ordinateur à la synthèse, en passant par le traitement, l'aide à la composition, l'analyse assistée, l'exécution, etc., et qui seront examinés de manière très circonstanciée, au travers de nombreux exemples.

Enfin, il y aura un séminaire de composition au cours duquel des compositeurs viendront présenter certains de leurs travaux, expliquer les enjeux esthétiques, les problèmes auxquels ils ont été confrontés, les éléments qui ont bien ou n'ont pas fonctionné, et éventuellement discuter des perspectives.

Ceci constituera la première partie : la formation théorique. La seconde partie, qui durera deux mois, sera consacrée à la réalisation d'un projet personnel. Les compositeurs s'y retrouveront "seuls", le temps de réaliser une série d'esquisses ou une petite pièce montrée dans leur atelier, pendant l'automne suivant, avec, éventuellement, la participation de l'Ensemble Intercontemporain.

Quels sont les rapports entre l'IRCAM et les conservatoires nationaux ?

Je m'occupe déjà, dans le cadre de la classe de composition du Conservatoire de Paris (1), d'un cours d'initiation à l'informatique musicale, réparti sur quasiment toute la durée des cours de composition, soit trois ans. Le Cursus de l'IRCAM a pour objectif de continuer cet enseignement, car, compte tenu des contraintes spécifiques du Conservatoire - contraintes d'occupation principalement -, le cours d'initiation ne peut rester qu'à un certain niveau. En fait, le Cursus d'informatique peut être l'un des débouchés d'une filière créée par le conservatoire : le troisième cycle.

Au CNSM de Lyon, les personnes qui s'occupent de la partie informatique musicale de la classe "Sonus" sont issus de l'IRCAM (Philippe Manoury). Une des cibles très originale de cette classe est de donner une base musicale plus large à des gens qui sont venus à l'informatique musicale - ou à la musique électronique - de manière un peu autodidacte, par des filières "non normalisées".

A l'issue du Cursus d'informatique musicale, les compositeurs peuvent repartir dans la nature, mais on peut aussi imaginer différentes manières de les insérer dans le tissu de l'IRCAM. Ainsi, ils pourraient faire partie de l'équipe des tuteurs, des assistants musicaux, et travailler pendant un an ou deux avec les compositeurs, afin de les aider à réaliser leurs projets. Ils pourraient aussi s'insérer dans l'équipe de recherche musicale et devenir des chercheurs travaillant sur un projet particulier. Il y a donc différentes stratégies d'insertion - ou de prolongation d'un séjour - à l'intérieur de l'IRCAM. Pour certains compositeurs, il n'est pas non plus impossible, de trouver, dans le cadre d'une relation assez "large", une forme plus légère et informelle d'accord qui leur permette de poursuivre, en pointillé, un projet personnel.

Quelle est l'attitude des compositeurs face aux outils informatiques ?

Chez à peu près tout le monde, il y a une motivation et un intérêt. Puis, suivant les personnalités, les réactions se diversifient. Dans certains cas, on observe une méfiance qui peut aller jusqu'à un désintérêt total, un rejet, car la volonté d'accepter l'investissement nécessaire est absente.

Cependant, dans la plus grande majorité des cas de figure, les compositeurs ont la volonté d'avoir accès à tous ces outils. Mais l'investissement consenti reste quand même très circonstancié, et pour cette catégorie d'utilisateurs, il faut que l'investissement en matière de connaissance, d'apprentissage, soit le plus court possible ; la distance entre la mise en oeuvre et le résultat doit aussi être la plus courte possible. Les outils sont assimilés et pensés dans la continuité des outils traditionnels, ils font partie de la palette, pourvu qu'une contrainte technologique ne vienne pas "refroidir" les utilisateurs dans leur appréhension.

Enfin, il y a quelques acharnés qui n'ont peur de rien et sont prêts à aller très loin dans la maîtrise approfondie du fonctionnement de la machine. Curieusement, ces derniers sont de plus en plus rares. Cependant, il faut apporter une précision : nous traitons une clientèle "triée sur le volet", qui, globalement, a suivi une formation plutôt traditionnelle et classique. Cette situation est différente de celle des débuts de l'IRCAM, il y a une quinzaine d'années. A cette époque, les personnes impliquées dans la musique électronique étaient principalement des spécialistes. Ce n'est plus du tout le contexte d'aujourd'hui, et nous payons maintenant le prix de ce degré d'engagement antérieur.

Constatez-vous une différence d'attitude chez les compositeurs les plus jeunes ?

Dans la génération montante, on constate une meilleure maîtrise de base, une meilleure disponibilité, et, en même temps, une impatience assez largement non-constructive. Cela veut dire que les jeunes compositeurs ne veulent pas faire d'investissements démesurés - des investissements temporels, dirais-je. Pour eux, il est important d'aller tout de suite à l'essentiel, sans s'encombrer de choses superflues, et cette attitude pose des problèmes terribles. Toute une génération d'outils, tel qu'UNIX par exemple, devient impensable et ne les intéresse absolument pas : il faut que tout fonctionne avec une interface de type Macintosh ou ST ! Des familles de problèmes, tout à fait intéressantes, deviennent, de ce fait, inabordables, car historiquement développées dans des contextes où les utilitaires de ce type n'existaient pas.

Pour les mêmes raisons, la grande majorité de ces jeunes compositeurs, n'est pas capable de considérer la relation temps réel/temps différé : il faut que le système réagisse en temps réel, afin d'avoir une interaction immédiate. Toute une famille d'outils et de préoccupations disparaissent complètement devant une telle attitude.

Faut-il alors adapter les outils ?

Les concepteurs de système, aussi bien à l'intérieur qu'à l'extérieur de l'IRCAM, doivent relever un défi : trouver des solutions de plus en plus efficaces, prenant les compositeurs dans le sens de leur motivation, mais aussi dans le sens de leur intuition. Jusqu'à présent, on ne partait pas de ces prémices-là, on essayait de définir un espace de possibles et on en déterminait l'accès, de manière très dépendante de l'état de la maîtrise des techniques. Maintenant, il va falloir se préoccuper de l'angle d'attaque, de la manière dont ces outils stimulent - ou ne stimulent pas - directement le compositeur, en regard de son apprentissage traditionnel. Par rapport au passé, le changement de philosophie est assez important.

Auparavant, le seuil critique de la maîtrise des moyens n'ayant pas été atteint, il était impossible de prendre ces exigences en considération. Désormais, les machines sont plus rapides, la technologie d'interface graphique rend l'interaction globalement plus facile - en tout cas pour certains types de problèmes - et il est possible de commencer à considérer les problèmes sous cet angle. Je pense, par exemple, qu'on va avoir de plus en plus d'interfaces, en terme de notation traditionnelle, intégrées dans un contrôle d'une autre nature.

Quels sont les types de problèmes les plus délicats à résoudre ?

La synthèse de très haute qualité - issue de l'expérience *Chant*, *Formes*, etc., et maintenant des modèles physiques - reste difficile à réaliser en temps réel, et quand on y parvient, la modélisation n'est pas possible sous une forme polyphonique. Mais cet aspect quantitatif, lié à la technologie, peut se résoudre aisément avec des machines qui "moulinent" plus vite.

Les problèmes d'ordre qualitatif sont beaucoup plus complexes, car la qualité de l'interaction avec la machine revêt toutes sortes de formes. En général, lorsqu'un compositeur souhaite faire quelque chose, on peut toujours, d'une manière *ad hoc*, trouver une solution. Par contre, les conflits entre généra-

lité et spécificité sont très délicats à maîtriser. Une solution, bonne pour tel compositeur, dans telle pièce et à tel moment, peut soulever des problèmes gigantesques dès qu'il s'agit de déborder du cadre fixé. De tels problèmes, liés à la nature du passage de l'idée du compositeur à sa réalisation, se manifestent principalement au travers des systèmes d'interface et de langage.

Quelles solutions entrevoyez-vous ?

Dernièrement, nous avons accompli d'importants progrès avec le programme *Patcher* (3), qui est fondé sur le paradigme des langages de programmation visuelle, et orienté vers l'aspect "performance". Il marque un effort dans le sens d'une programmation modulaire reposant sur un système de bibliothèques, car à partir d'un certain nombre de cas de figure prédéterminés, le compositeur conserve la maîtrise de l'assemblage. La réalisation des modules et la programmation interne des entités graphiques que l'on raccorde les unes aux autres est réservée à des utilisateurs spécialisés.

Cependant nous ne sommes pas encore parvenu au stade où un compositeur peut, dans un premier temps, décrire son idée d'une manière extrêmement intuitive, et obtenir ensuite le résultat. L'enjeu le plus considérable dans les années à venir est pourtant là. Les recherches vont donc porter sur la découverte des formalismes les plus directement connectés à la manière dont l'imagination des compositeurs fonctionne. Il y a une telle hétérogénéité des formations, des idées, de la manière de penser, que de trouver des outils offrant les biais les plus courts possibles entre l'idée et le résultat, devient un projet extraordinairement complexe.

Aujourd'hui les compositeurs disposent de boîtes à outils, qu'ils peuvent plus ou moins individualiser, avec lesquelles ils peuvent réaliser leurs applications, et arriver à construire leur propre univers, éventuellement aidés par un assistant. Le prochain cap sera de pouvoir façonner leur univers d'une manière naturel-

FORMATION

Dans le domaine de la formation aux techniques informatiques, l'IRCAM déploie ses activités pédagogiques dans trois directions complémentaires, correspondant à des cycles d'étude de plus ou moins longue durée :
- initiation ;
- spécialisation ;
- cursus complet.

Ces études sont avant tout destinées à compléter la formation des compositeurs issus des Conservatoires Nationaux, en leur apportant les connaissances théoriques et pratiques nécessaires à la mise en oeuvre de systèmes informatiques comparables à ceux qui équipent les studios de l'IRCAM.

Cours d'Initiation à l'Informatique Musicale

Il est spécifiquement réservé aux étudiants du CNSM de Paris, et s'articule sur un enseignement théorique hebdomadaire (séances de 3 heures) assorti de travaux pratiques effectués en studio.

Stages spécialisés

Ils s'adressent aux compositeurs qui souhaitent acquérir, dans l'optique d'une production professionnelle, les bases pratiques des technologies musicales reposant sur les systèmes d'information. Ils se déroulent sur une courte période d'enseignement intensif (2 à 5 jours complets), et sont centrés sur un thème unique :
- acoustique et psycho-acoustique ;
- analyse, synthèse et traitement des sons par ordinateur ;
- technologie des systèmes MIDI ;
- composition assistée par ordinateur ; etc.

Cursus d'Informatique Musicale

Un enseignement complet, courant sur une année entière, est destiné à assurer la formation des compositeurs ayant achevé leurs études dans les conservatoires nationaux (ou les compositeurs étrangers issus d'établissements équivalents). Il se décompose en deux phases :
- formation pure,répartie sur une période de 25 semaines (octobre à mai), à raison de 4 séminaires hebdomadaires de 3 heures, au cours desquels les aspects théoriques et pratiques de l'informatique musicale sont examinés :
  - science et musique :
    - acoustique ;
    - psycho-acoustique ;
    - traitement du signal audio-numérique ;
    - méthodologies de recherche ; etc.
  - initiation à l'informatique et à la programmation ;
  - informatique musicale :
    - analyse, synthèse et traitement informatique des matériaux sonores ;
    - composition assistée par ordinateur ;
  - analyse et composition ;

  la partie pratique de cette formation étant exécutée sur des stations d'esquisse (station musicale IRCAM ou systèmes personnels de type Macintosh) ;
- réalisation de projet individuel, dans le cadre d'un environnement de studio informatique, dont le résultat est programmé au cours des ateliers-concerts de la saison qui suit. Il est réparti sur 2 mois pleins (juin et juillet).

D'autre part, dans le contexte d'une formation à la musicologie du XXe siècle, l'IRCAM organise plusieurs séminaires, dont un consacré aux nouvelles technologies. Au cours d'une quinzaine de séances sont abordés les thèmes suivants :
- histoire de l'évolution des technologies ;
- acoustique et psycho-acoustique ;
- synthèse et traitement des sons par ordinateur ;
- sonorisation et pratique de studio ;
- temps réel ;
- composition assistée par ordinateur ;
- perspectives des recherches musicales ; etc.

le, de le décrire en termes de contraintes extrêmement intuitives, ne serait-ce que grâce à l'expérience des mécanisme d'apprentissage.

Dans un contexte aussi complexe, les activités pédagogiques de l'IRCAM peuvent-elles couvrir tout le champ de l'informatique musicale ?

Pendant longtemps, le champ d'application de l'informatique musicale a été, presque exclusivement, le traitement de signal - principalement analyse et synthèse. Aujourd'hui on constate une fantastique explosion dans le domaine de la composition. Ceci est une conséquence de l'investissement fait autour du contrôle de la synthèse - à travers le programme *Esquisse* et la famille d'outils issue de *PréFormes* et de *Patchwork*. Ce domaine d'aide à la composition, qui découle directement des expériences menées dans le domaine de l'intelligence artificielle - et, à la limite, sert à faire de la musique purement instrumentale - est maintenant assez bien représenté à l'IRCAM.

Mais, les années passant, nous ne sommes plus sûrs d'arriver à créer des pôles de compétences sur tous les domaines diversifiés qui sont désormais impliqués. L'analyse assistée par ordinateur, qui peut être vue comme une autre face de l'aide à la composition, est, par exemple, un sujet sur lequel il faudra être présent, car il intéresse beaucoup les musiciens. Même sur les thèmes "traditionnels", nous avons de plus en plus de difficultés à être présents, car l'informatique musicale est maintenant devenue un marché. Un nombre de personnes considérable se trouve concerné par ce marché, et nous sommes obligés de constater qu'il nous est impossible de rivaliser avec certaines forces industrielles.

Cela a-t-il une incidence sur les relations qui lient le compositeur et l'IRCAM ?

Il faudrait que l'IRCAM reste une structure capable de s'adapter à ce que demandent les compositeurs. Mais aller jusqu'au bout de ce principe, peut nous conduire à des situations insupportables : faire une machine spécialisée pour chaque pièce, etc. D'autre part, l'expérience montre que les compositeurs sont déstabilisés s'ils n'ont pas à supporter un réseau de contraintes. Il s'agit donc de trouver un bon compromis.

Le thème de l'aide à la composition est un thème moteur car on peut construire des environnements d'aide à la composition assez sophistiqués sur une machine comme le Macintosh, donc utilisable chez soi. Il s'agit là d'une possibilité importante, dans le cadre des activités pédagogiques, car on peut "renvoyer" le compositeur chez lui ! Venir ici, travailler en studio avec un assistant, supporter toutes sortes de contraintes sociales - car l'IRCAM est un lieu où il y a beaucoup de passage et des pressions psychologiques très fortes -, force est de reconnaître que tout cela a un impact sur la qualité du travail. Si le compositeur peut réaliser chez lui la majeure partie de son travail, en possédant la maîtrise de son outil de production, il y a toutes les raisons de penser qu'il arrivera à s'épanouir un peu mieux.

Il faut maintenant arriver à axer la relation entre le compositeur et l'institution sur un rapport beaucoup plus sain, dans lequel la structure n'est utilisée que pour ce qui lui est spécifique et très bien circonscrit, c'est-à-dire des outils de synthèse très sophistiqués. D'ici quelque temps il sera possible de réaliser l'essentiel de la modélisation chez soi, même et y compris toute la synthèse sous une forme simplifiée. Venir à l'IRCAM doit donc être, du point de vue des ressources matérielles spécifiques, une manière d'appuyer sur l'accélérateur. Mais il faut une continuité maximum entre la manière de travailler chez soi et la manière de travailler à l'IRCAM.

Il y a une part sociale dans la pratique de la musique et, même avec une machine faisant l'essentiel de la synthèse, il faut quand même bien aller dans la salle de concert ou le studio. Même s'il faut renvoyer le compositeur chez lui, je ne pense pas qu'il faille absolument supprimer le rôle social, car il ne faut pas perdre le rapport à la structure. Il faut simplement se centrer sur les choses vitales, à savoir tous les aspects sociaux. Ceux-ci passent par les rapports entre le compositeur et l'instrumentiste, l'instrument et la machine, et également par le rapport entre le compositeur et le chercheur. Ce qui est recherché dans ces rapports, ce ne sont plus des machines super-performantes, mais des des compétences et des connaissances : le compositeur vient poser un problème pour avoir un avis, une solution, des esquisses, une palette de solutions. Venir consulter des spécialistes, leur exposer des problèmes théoriques complexes qui nécessitent un bagage en mathématique, en physique, ou encore en psycho-acoustique est une attitude véritablement constructive.

L'IRCAM doit être là pour faire la conjonction de toutes ces compétences, plus que pour des ressources matérielles et logicielles, et ce rôle social me paraît extrêmement important.

Propos recueillis par
Daniel Fournier

(1) Conservatoire National Supérieur de Musique de Paris.

(2) *Macintosh Programming Workbench.*

(3) Voir, dans ce numéro, l'article sur la station musicale IRCAM.

LE SYSTEME 4X

Élaboré en plusieurs étapes, marqué par des prototypes successifs (4A, 4B, 4C), le système de traitement de signal numérique 4X conserve, près de dix années après sa réalisation, des capacités exceptionnelles. Il constitue désormais un modèle de référence pour toutes les machines de traitement du son numérique.

Doté de performances très élevées (200 millions d'opérations/seconde), le système 4X est dédié au calcul des échantillons numériques permettant de réaliser tous les types de synthèse algorithmique actuellement connus :
- additive ;
- soustractive et filtrage numérique ;
- modulation de fréquence ;
- échantillonnage ;

ainsi qu'une grande variété de traitements typiques du son :
- modulation en anneaux, harmonisation ;
- réverbération, écho, phasing ;
- analyse fréquentielle (FFT, corrélations, suivi de hauteur), etc.

Ses caractéristiques principales correspondent aux exigences actuelles de l'audionumérique professionnelle :
- données représentées sur 16 bits ;
- dynamique atteignant 96 dB ;
- bande passante de 20 kHz.

ARCHITECTURE

Conçu comme périphérique d'un ordinateur central (pourvu d'une interface utilisateur et de périphériques de stockage), le système 4X est un système modulaire, reconfigurable en temps réel par microprogrammation, composé de :
- 1 à 8 cartes de traitement du signal (4U) ;
- 1 carte d'interface avec la machine hôte ;
- 1 carte de contrôle pour l'acquisition et la restitution des données numériques (convertisseurs A/N et N/A).

Chaque carte 4U peut travailler à sa propre fréquence d'échantillonnage (variable de 16 à 512 kHz), afin d'obtenir une bande passante s'étendant de 8 à 256 kHz.
Une carte 4U comporte deux unités de calcul :
- arithmétique et logique ;
- multiplication.

Les calculs arithmétiques sont effectués en virgule fixe, sur des données codées sur 24 bits, selon un cycle de base de 60 ns (16 MHz).

Plusieurs mémoires internes complètent la carte 4U :
- mémoire de fonctions, 64 Kmots en codes opérations de 16 bits ;
- mémoires de données, 24 Ko en données de 24 bits ;
- mémoires d'adresses ;
- mémoire de microprogramme, 1024 instructions codées sur 48 bits.

Les instructions des différentes unités, décomposées en étapes s'exécutant dans un pipeline, sont effectuées en parallèle.

La carte de contrôle du système 4X est pourvue de :
- 256 temporisateurs programmables, générant un signal d'interruption sur une ligne reliée au calculateur hôte ;
- 2 mémoires tampons, pour lire/écrire les données sur disque ou en mémoire du calculateur hôte ;
- un module de gestion du bus interne du système 4X ;
- un module de gestion des E/S.

PROGRAMMATION

La programmation d'un système 4X peut s'effectuer à deux niveaux : assembleur et langage évolué compilé.

Assembleur-optimiseur
Un assembleur décrivant les opérations élémentaires (transferts registre/registre, mémoire de données/registre, lecture/écriture en mémoire de fonctions, etc.) a été couplé à un optimiseur qui assure la gestion du parallélisme des instructions. Le parallélisme est en fait totalement transparent pour le programmeur, qui peut concevoir la

structure de son programme de manière séquentielle. Cet assembleur peut être mis en oeuvre aussi bien en mode compilé qu'en mode interprété.

Compilateur 4X
Développé par Andrex Gerzso, le *4X Patch Language* a pour but principal de faciliter l'élaboration et la programmation des algorithmes d'analyse et de synthèse des sons numériques. Il permet d'utiliser des identificateurs symboliques pour référencer les données stockées en mémoire ou dans les registres internes des cartes 4U.

La syntaxe du langage a été volontairement simplifiée. Elle répond à la forme :

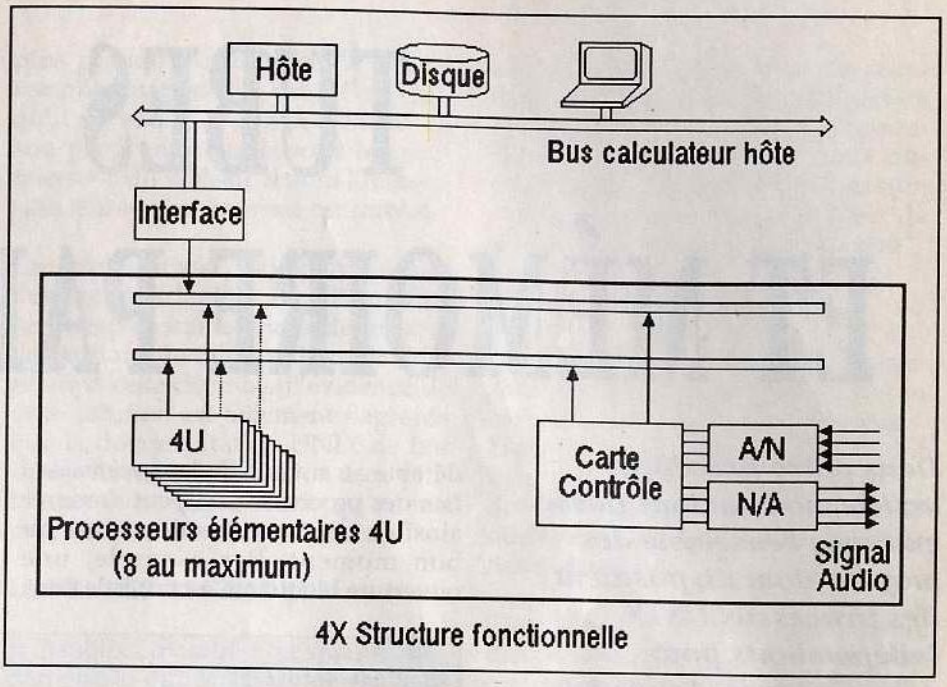
```
(sorties) =
nom(vars_partitions|vars_entrées)
(
(déclaration_de_variable)
def {
    variable = valeur ;
    ...
    }
expressions ;
copy n_fois (expressions)
)
```

La commande 'copy' est utilisée pour signifier la duplication du code, avec incrémentation d'adresse, dans le cas d'algorithmes récursifs.

Une variable peut être :
• simple (codée sur les 24 bits supérieurs d'un mot de 32 bits);
• vectorisée - la dimension est indiquée entre [], en partant de l'indice 0, comme dans le langage C ;
• de dimension indéterminée.
Elle peut également être affectée d'un facteur d'échelle utilisé lors de son initialisation.

Les convertisseurs N/A et A/N sont représentés par leurs indices dans le vecteur qui les désigne :
`in[0] ... in[15]` pour les A/N ;
`out[0] ... out[15]` pour les N/A.

Le langage reconnaît un certain nombre d'opérations simples :
• addition et soustraction, avec option de contrôle du dépassement de capacité ;
• fonctions logiques (AND, OR, XOR) ;
• décalages arithmétiques et logiques à droite et à gauche ;



4X Structure fonctionnelle

• lecture et écriture en mémoire de fonction (RFUN, WFUN).

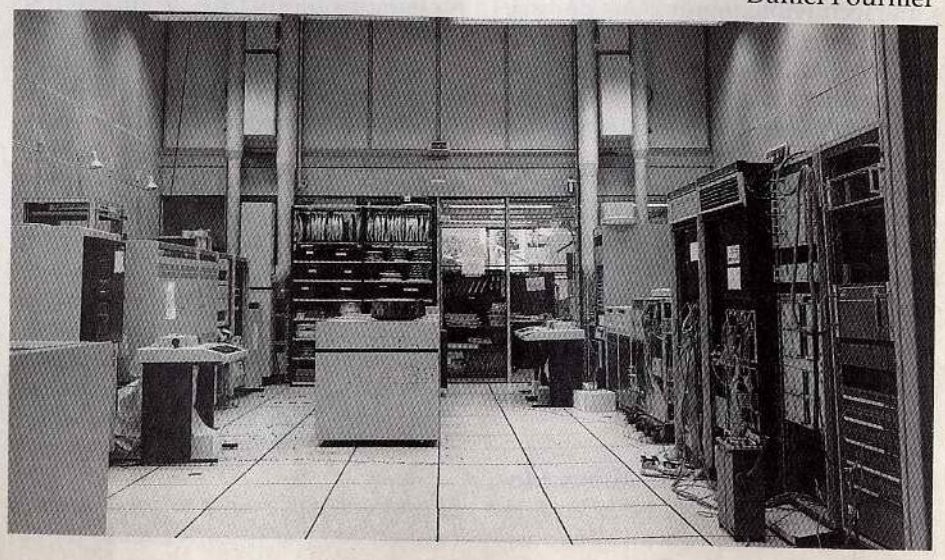
Le compilateur 4X possède un optimiseur de code assembleur généré à la suite de l'analyse lexicale, syntaxique et sémantique. Ce code assembleur est ensuite lui-même optimisé, comme nous l'avons vu plus haut.

LA STATION 4X

Le système 4X forme l'épine dorsale des quatre stations musicales 4X dont dispose l'IRCAM. Chacune comprend :
• un poste de développement fonctionnant sous UNIX, et qui permet de mettre en oeuvre le compilateur et l'assembleur ;
• un calculateur temps réel, à base de microprocesseur MC68000, entouré de périphériques de contrôle, pupitre de commandes analogiques, clavier musical, interfaces MIDI, reliés par un bus VME ;
• un système 4X assorti de convertisseurs A/N et N/A.

A terme, ces stations 4X, sur lesquelles de nombreuses oeuvres commandées par l'IRCAM ont vu le jour, devraient céder la place à la nouvelle génération de machines, celle de la nouvelle station musicale basée sur le NeXT. L'empreinte du système 4X n'en demeurera pas moins gravée dans l'histoire musicale de la fin du vingtième siècle.

Daniel Fournier



TUBES ET MÉMOIRE PARTAGÉE

Dans notre précédent article, nous avions passé en revue l'ensemble des moyens dont disposaient des processus UNIX indépendants pour communiquer. Cet article complète certains points de détails sur les tubes, et présente le concept de mémoire partagée, avec ses contraintes d'exclusion mutuelle.

OUVERTURE NON BLOQUANTE DE TUBES BLOQUANTS

Les tubes sont par défaut en mode dit *bloquant*, c'est-à-dire que la lecture, l'écriture, et l'ouverture peuvent mettre le programme en sommeil (l'ôter de la pile des processus actifs, qui accèdent périodiquement au processeur central) :

- une lecture bloquante dans un tube vide endort le processus - il ne sera réveillé que lorsqu'un autre processus aura écrit quelque chose dedans ;
- une écriture bloquante dans un tube plein endort le processus - il ne peut être réveillé que lorsqu'un autre processus a suffisamment vidé le tube (lu dans le tube) pour qu'il y ait la place d'écrire ;
- une ouverture d'un tube dans lequel aucun processus ne peut écrire endort le programme jusqu'à ce qu'un autre processus ouvre le même tube en écriture ;

Il semble donc que les tubes bloquants n'apportent que des ennuis. Mais ils présentent un intérêt considérable au niveau de la *synchronisation* des processus : on peut s'assurer ainsi qu'un processus démarrera au bon moment. Par exemple, une ouverture bloquante est cruciale dans le cas où le processus qui doit lire dans le tube est lancé *avant* le processus qui doit y écrire : si l'ouverture n'était pas bloquante, le processus lecteur se mettrait à lire avant que le processus qui écrit soit lancé, trouverait le caractère FIN_DE_FICHIER, et s'arrêterait prématurément.

Malheureusement, si les tubes à lecture et écriture bloquantes sont utiles, on aimerait parfois que leur ouverture ne le soit pas, ce qui n'est pas possible avec les tubes primitifs d'UNIX (on a le choix entre les tubes bloquants et non bloquants, mais pas au panachage des propriétés). On est donc obligé de recourir à une astuce : le lancement de faux processus.

Si on veut par exemple que l'ouverture en lecture d'un tube bloquant ne soit pas bloquante, il suffit de lancer un faux processus, qui va "faire semblant" d'écrire dans le tube (pour réveiller le processus utile) et mourir juste après :

```
#include <stdio.h>
/* on suppose qu'il existe déjà un tube nommé de nom :     */
char     *NomDuTube = "tube_nomme";

main ()
{
FILE     *PIn;

    if (fork() == 0)
    {          /* processus bidon  */
    FILE       *PBidon;

         PBidon = fopen ( NomDuTube , "w" );      /* réveil du père */
         exit(0);                    /* ne sert qu'à réveiller le père */
    }
    PIn = fopen ( NomDuTube , "r" );
    /* ... le processus père continue presque instantanément ...  */
}
```

Cette astuce ne peut pas, par contre, fonctionner directement de cette manière pour ouvrir un tube bloquant *en écriture*, car un processus qui peut écrire dans un tube qui ne peut être lu par aucun autre processus reçoit un signal SIGPIPE, qui met radicalement fin à ses jours. Pour éviter cela, on installe généralement une temporisation `sleep` juste avant l'ordre `exit`, ce qui laisse le temps à un autre processus d'ouvrir le tube en lecture :

```
main ()
{
FILE     *PIn,*POut;

    if (fork() == 0)
    {
    FILE       *PBidon;
```

```
        PBidon = fopen ( NomDuTube , "r" );
        sleep ( 1000 );
        exit (0);
    }
    POut = fopen ( NomDuTube , "w" );
    PIn  = fopen ( NomDuTube , "r" );
    /* suite du programme, qui dispose */
    /* d'une variable FIFO personnelle */
    /* ... */
}
```

Le premier `fopen` du père est réveillé par le `fopen` du fils. Si par malheur le fils mourait avant le deuxième `fopen` du père, le père recevrait un SIGPIPE. C'est pourquoi la temporisation `sleep` assure que le deuxième `fopen` du père est bien effectué avant la mort du fils.

TAILLE DES TUBES

Suivant les versions d'UNIX, la taille des tubes (c'est-à-dire le nombre de caractères qui peuvent être écrits avant que l'écriture soit bloquante), peut varier considérablement. Cependant, elle est toujours faible, en général inférieure à 10 Ko.

Cette caractéristique des tubes peut poser un problème : comment fait-on pour passer, à travers ce tube, un gros tableau ? Cette question peut paraître à première vue simple à résoudre : puisque l'écriture est bloquante, même si on écrit un gros tableau, l'écriture s'endormira le temps que le programme qui lit vide suffisamment le tube. De même, si on lit un gros tableau, l'ordre `read` se bloquera périodiquement, le temps que le programme qui écrit remplisse à nouveau le tube.

Malheureusement, la fonction `read()` (ou `write()`) ne vérifie le caractère bloquant du tube qu'une seule fois : au moment où elle est appelée. La conséquence est qu'un ordre `read` chargé de lire 2 Mo commencera par tester si le tube contient quelque chose, puis lira brutalement 2Mo, et enfin renverra un code d'erreur signifiant que tout s'est déroulé normalement !

Voyons maintenant ce qui se passe du point de vue du programmeur : il a beau truffer son programme de tests, UNIX lui affirme que tout se passe correctement, mais pourtant la plus grande partie de ses 2 Mo est remplie du caractère nul ! Pour peu qu'il ne vérifie le comportement de son programme que pour les premiers Ko du tableau, il aura l'impression que son programme est correct.

La solution est évidemment de découper la lecture de 2 Mo d'un seul coup en une myriade de lectures de portions de tableau de taille inférieure à celle du tube. L'évidence de cette solution est tellement flagrante que la documentation UNIX ne fait jamais la moindre allusion à ce problème, voire laisse croire qu'aucun problème ne peut se produire...

LES TUBES AMÉLIORÉS

Il manque aux tubes une qualité fondamentale qui pourrait encore augmenter considérablement leur utilité.

Dans un programme classique utilisant les tubes, la procédure chargée de la lecture est obligée de passer son temps à attendre que quelque chose veuille bien se présenter, que ce soit par une lecture bloquante ou par une scrutation périodique. Il serait bien sûr beaucoup plus facile d'être *prévenu* lorsque le tube contient quelque chose. Or, les interfaces de programmation récentes (GEM, Windows, X Window, etc.) sont basées sur ce principe pour une grande partie des événements qu'elles doivent gérer (clavier, souris, etc.).

Il est donc naturel que certaines d'entre elles proposent au programmeur des tubes améliorés, capables de prévenir lorsqu'un autre processus écrit dans le tube, c'est-à-dire déclencher automatiquement une procédure déclarée par le programmeur comme pouvant gérer cet événement. Cela se révèle incroyablement plus utile que ça en a l'air à première vue. En fait, il faut tenter de transposer un programme utilisant massivement ce concept vers une autre interface de programmation pour se rendre à quel point c'est un outil puissant. En effet, il combine le grand débit de données des tubes normaux avec le principe d'interruption logicielle des signaux.

Cet article ne présente pas d'exemple de programme utilisant ce genre de tube, car il faudrait exposer un grand nombre de définitions spécifiques de l'interface de programmation considérée. Cependant, pour ceux qui connaissent Sunview, la fonction `notify_set_input_func()` permet de déclarer ces "tubes qui préviennent".

LA MÉMOIRE PARTAGÉE

Les tubes sont des outils très intéressants par leur simplicité de mise en oeuvre et par leurs performances. Mais il est un domaine où ils sont très mal adaptés : c'est lorsqu'ils doivent communiquer à un grand nombre de processus des informations identiques.

C'est le cas lorsqu'on souhaite avoir des *variables globales* à plusieurs processus. Au-delà de 4 ou 5 processus, il faut un nombre de tubes plutôt grand, et cela devient très lourd à gérer. De plus, tous les ordres d'écriture doivent être dupliqués : un message par processus à prévenir.

Rappelons simplement que les différents processus partagent les mêmes adresses en mémoire, ce qui fait qu'ils ne peuvent absolument pas communiquer par des variables globales à adresse fixe, comme c'est le cas sur micro (cf. l'article précédent, qui montre que lors d'un `fork`, le processus fils, bien qu'étant exactement aux mêmes adresses que son père, disposait d'une copie *privée* de l'image mémoire du père, dans laquelle les modifications ne sont pas reportées vers l'autre processus).

Pour résoudre ce genre de difficulté, on va appliquer le même principe que lorsqu'il faut rendre des variables globales à plusieurs procédures dans un programme classique. Lorsqu'il y a trop de variables communes dans un groupe de procédures, on les rend globales : en Pascal ou en C, elles deviennent globales au fichier (catégorie `static` du C), ou complètement globales (catégorie `extern` du C). En Modula-2 ou en Ada, on peut ajuster plus finement la portée des variables.

Le principe retenu par les primitives de mémoire partagée d'UNIX System V est assez différent : la visibilité d'une zone de variables est défi-

nie par une *clé*. Chaque processus possédant la clé d'une zone de mémoire partagée peut y accéder, et il faut bien sûr absolument disposer de la clé pour y accéder.

La zone de mémoire partagée est accessible par un pointeur, ce qui fait que son comportement est exactement identique aux zones de mémoire privées, la différence étant que les autres processus accédant à la même zone de mémoire peuvent modifier son contenu à tout moment et sans prévenir.

Les rois de l'assembleur M68000 et autres bidouilleurs de variables système de leur ST se posent déjà la question cruciale : comment cela se passe-t-il si deux processus écrivent en même temps dans la même case mémoire ? Que va lire un processus au moment où un autre est en train d'écrire ?

Sur micro, on a la bonne habitude d'interdire les interruptions lorsqu'on accède à ce genre de variable. C'est strictement impossible sous UNIX : aucun processus (et surtout pas un processus lancé par un utilisateur) ne peut s'attribuer l'intégralité de la puissance du processeur. Ce qui fait que le programmeur n'a aucune garantie quant au temps que va mettre son processus à effectuer une portion donnée de code (il faut pour cela des systèmes d'exploitation dits *temps réel* ; il existe d'ailleurs certaines versions d'UNIX capables de ce genre de performance). Or, on ne doit pas interdire une interruption pour un temps trop important.

Enfin, sur une machine UNIX, les interruptions ont une fonction infiniment plus vitale que sur micro, sans parler de la complexité de ce qu'elles doivent gérer. Il faut donc mettre en place un système d'exclusion mutuelle. On verra dans un prochain article comment les *sémaphores* permettent de résoudre cette difficulté. Pour l'instant, regardons comment on utilise une mémoire partagée.

La première action est de définir une clé, qui préservera la confidentialité de la zone de mémoire vis-à-vis des processus qui ne doivent pas y accéder. Cette clé est tout simplement un nombre entier déguisé en type `key_t` en C. On fait alors appel à la fonction `shmget` :

```
int shmget (cle, taille, flag)
int         taille,flag;
key_t       cle;
```

Cette fonction est chargée de renvoyer l'identificateur de la zone de mémoire partagée, ce qui permet d'appeler d'autres fonctions de gestion de ces zones de mémoire (un peu comme le `VDIHandle` sous GEM permet d'accéder aux fonctions VDI).

Ensuite, on *attache* le segment de mémoire partagée à une adresse gérée par le programmeur. En général, cette adresse est considérée comme un pointeur vers une structure, ce qui permet de considérer le segment comme une simple variable structurée :

```
(char *) shmat(ident, adresse, flag)
(Structure *) adresse;
int           ident,flag;
```

Puis on peut consulter les caractéristiques de ce segment par la fonction `shmctl()`, qui est surtout utile aux programmes qui ne créent pas le segment, mais se contente de l'attacher à un de leurs pointeurs. On peut ainsi avoir accès aux informations suivantes :

- droits d'accès au segment (un peu comme pour un fichier)
- taille en octets
- PID du processus ayant effectué la dernière opération
- date du dernier attachement
- date du dernier détachement
- date du dernier changement

```
int shmctl (ident, operation,
            descripteur)
int         ident, operation;
shmid_ds    *descripteur;
```

L'opération peut être l'une des trois suivantes :

- IPC_STAT : lecture de la structure `shmid_ds`, qui est placée dans `descripteur`. Cette structure rassemble les informations listées ci-dessus ;
- IPC_SET : cette fois, c'est le contenu de `descripteur` qui est écrit dans la structure `shmid_ds` de la zone de mémoire ;
- IPC_RMID : détruit le segment de mémoire. Cette opération est indispensable, car sinon le segment de mémoire est maintenu jusqu'à la prochaine réinitialisation de la machine, ce qui peut être gênant si le segment est important ou si beaucoup de programmes omettent de détruire leurs segments. Rappelons qu'une machine UNIX est réinitialisée beaucoup moins souvent qu'un micro, car elle fait généralement partie d'un réseau, et son arrêt, même momentané, constitue une dégradation de la qualité globale des performances du réseau.

Enfin, si on ne souhaite pas détruire le segment, pour pouvoir le retrouver intact lors d'une future utilisation du logiciel, ou pour ne pas perturber d'autres programmes qui l'utilisent simultanément, on peut se contenter de détacher le segment par la fonction `shmdt` :

```
int shmdt ( adresse )
(Structure *) adresse;
```

On verra comment synchroniser les accès à la mémoire partagée lors d'un prochain article. Mais ces précautions sont inutiles dans un cas bien précis : si on souhaite lire ou écrire une variable de 1, 2 ou 4 octets. En effet, bien que cela paraisse un peu "sale", on peut compter sur le fait que les ordinateurs récents peuvent lire ou écrire des variables de cette taille *en une seule instruction du microprocesseur*. Ces instructions sont bien sûr non interruptibles, ce qui assure l'exclusion mutuelle. Bien que cela semble à première vue une astuce fort peu recommandable, elle est assez utilisée, car la lourdeur de l'utilisation des sémaphores, même "recarrossés" par des fonctions plus conviviales, fait éviter au maximum leur utilisation pour des opérations aussi simples.

Marc Lussin

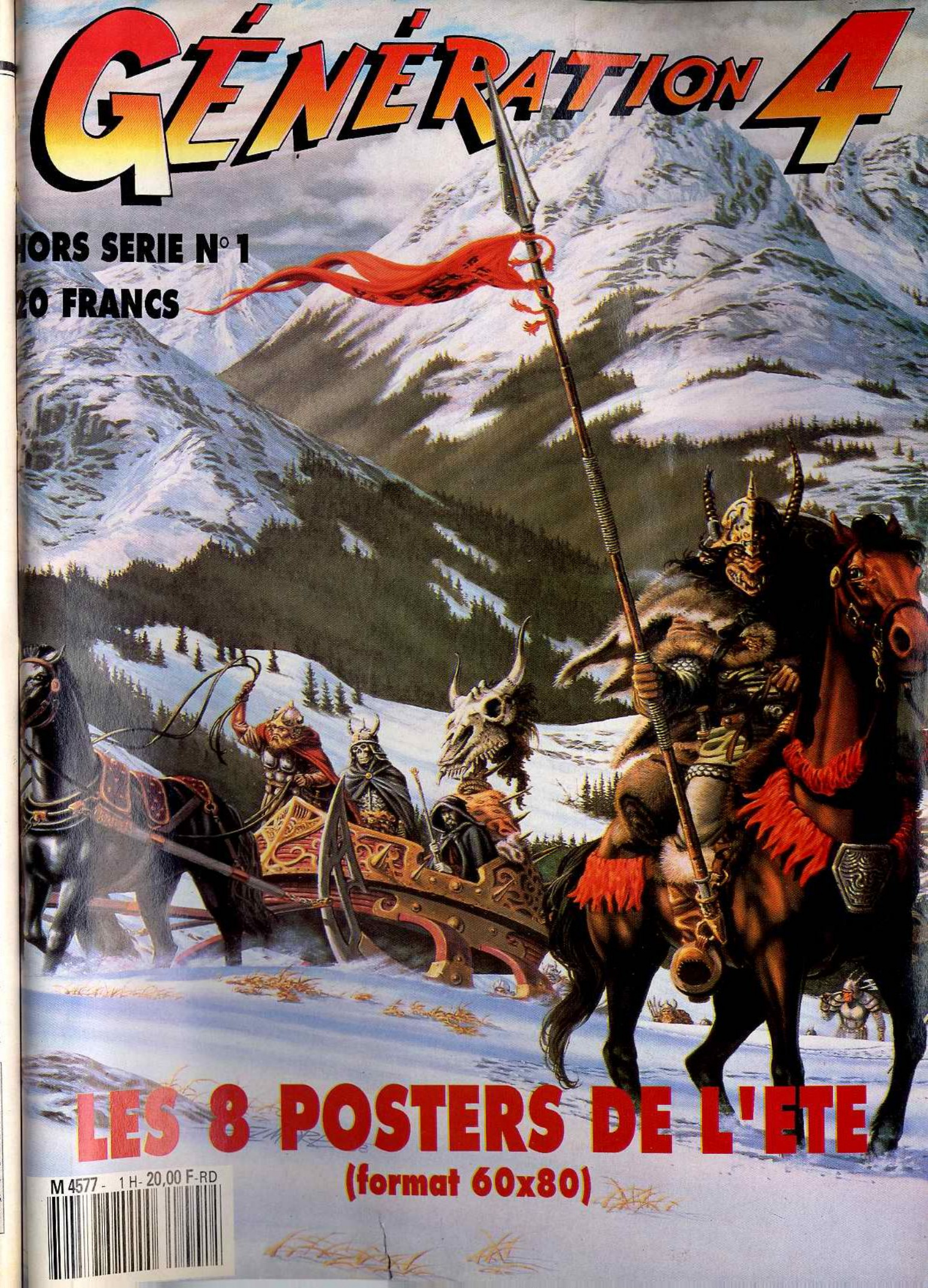
Prochainement...

Nous examinerons en détail les sémaphores, et les différentes manières de les "cuisiner" (il y en a plusieurs !). Pour exercice, on verra qu'on peut simuler les tubes avec les sémaphores et la mémoire partagée, qu'on peut simuler les sémaphores avec 3 tubes ou un tube et des signaux, etc.

GÉNÉRATION 4

HORS SERIE N° 1

20 FRANCS

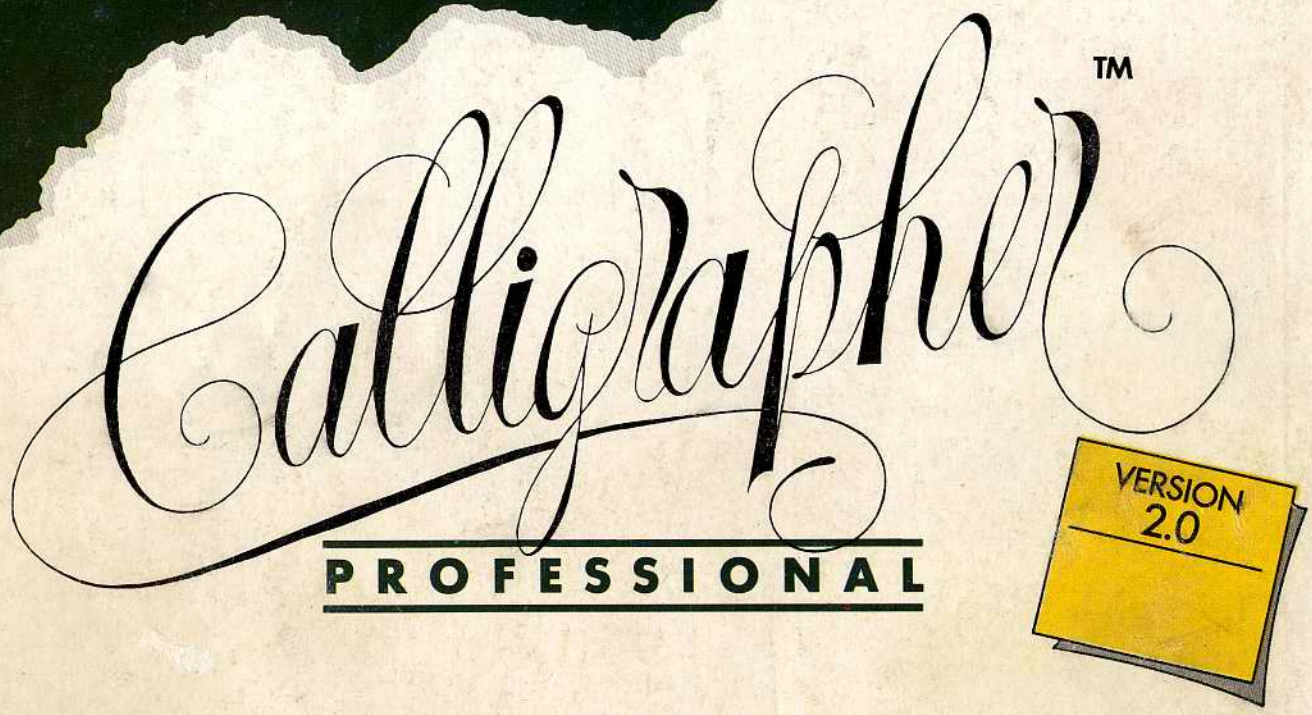


LES 8 POSTERS DE L'ETE

(format 60x80)

M 4577 - 1 H - 20,00 F-RD

SOUS LES PAVES LA PAGE.



IL existe aujourd'hui une nouvelle liberté d'expression. Pour la première fois se trouvent réunis au sein d'un traitement de textes :

- la puissance (glossaire universel, dictionnaire, processeur d'idées, notes et commentaires, feuilles de style, gestion de chapitres inter-documents, publipostage, outils graphiques).
- la souplesse (multi-colonnage réel, mode graphique haute résolution WYSIWYG ou mode texte rapide, polices vectorielles ou polices GDOS®, import/export de nombreux formats de fichiers textes ou graphiques, option réseau)
- la facilité d'utilisation (installation automatisée, création instantanée de tableaux ou de formules mathématiques, prévisualisation du document, interface utilisateur intuitive).

Découvrez la liberté d'écrire.

Upgrade EDITIONS

28/30, rue Coriolis, 77012 Paris. Tél. (1) 43 44 78 88. Fax (1) 43 44 90 96

Les marques citées sont déposées par leurs propriétaires respectifs.
© 1990 ECLECTRON SA, Suisse, et Upgrade Editions, Paris.

CALLIGRAPHER™ PROFESSIONAL, LE TRAITEMENT DE TEXTES POUR ATARI® ST